UNLEASHING THE POWER OF AMAZON WEB SERVICES WITH ACCENTURE

HARNESSING AWS BEYOND JUST A COMMODITY DATA CENTER
1. EXECUTIVE SUMMARY

First impressions can eventually get in the way of fully understanding a subject as important and complex as cloud computing. When the cloud first entered the consciousness of IT organizations around the world, cost-cutting through commoditization was foremost on people’s minds. Maybe that worked for a while, but it no longer does. Today, we see real problems arise when organizations try to drive a cloud migration leveraging Amazon Web Services (AWS) as just a commodity data center: “bill shock,” “operational complexity,” “reduced availability,” “under-performance,” “security creep,” “cloud sprawl,” “redundant software licenses” and “procurement perplexity.” One thing is for sure: Cloud is not about driving toward the lowest common denominator.

What AWS is really about is enabling a business to “pivot to the new.” AWS, when leveraged in its entirety, offers organizations the ability to **transform the way they consume and architect applications native to the cloud.**

Accenture AWS Business Group (AABG), working with clients and delivery teams from both Accenture and AWS, can report that leveraging AWS foundational services delivers a host of important benefits:

- Companies can save up to 56 percent on the bottom line for enterprise applications hosted on AWS.  

- Moving up the AWS stack and adopting managed services that automate delivery of AWS can reduce the cost to serve by nearly 80 percent.

- We see an improvement in the top line through accelerating time to deliver by 95 percent and improving the quality of the delivered workloads by 75 percent. The speed and frequency to deliver along with the quality improvement is directly correlated to employee effectiveness, resulting in improved customer satisfaction scores.

- Supporting our clients in adopting a fully cloud-native application architecture can result, on average, in reduced time to delivery by up to 90 percent and up to a staggering 99 percent reduction to the bottom line.

- The impact on security, resilience and compliance is significant, with overall reductions in outages and security-related incidents.

- AABG projects also see considerable reduction in the time to deliver. At one client we saw a 52 percent reduction in the effort to migrate to AWS.

An AWS migration journey that looks to decouple infrastructure services and embrace cloud-native services offers an opportunity to reinvigorate organizations both in terms of agility and business value delivered.

When considering the adoption of cloud-native architectures and the methods to support them, it can be daunting to know where to begin and where to commit investments. Without guidance, many companies may fail to achieve their desired IT and business outcomes. In this document, the Accenture AWS Business Group provides a prescriptive approach to ease the complexity of a company’s cloud journey. Leveraging the knowledge and experience gained across both Accenture and AWS, and in alignment with the “AWS Well Architected Framework,” we will describe the key components that define an optimized AWS migration approach. The content in this document will focus on how best to implement and adopt AWS native tooling, in conjunction with application and migration approaches, to achieve positive business outcomes.
2. INTRODUCTION

This document, “Unleashing the power of Amazon Web Services with Accenture,” outlines a prescriptive overview of the AWS services and application design patterns to consider when exploring an application migration approach to AWS. It also explores the various cloud-native methods and models that AABG recommends when planning to leverage AWS to increase business value and agility.

The main intent of the document is to provide a holistic picture on how best to consume AWS’s services and architecture patterns as an enabler for moving towards a cloud-native architecture. The value in rethinking the way in which applications and services are deployed on AWS can lead to a substantial improvement in return on investment (ROI).

This analysis begins by exploring the foundational thinking around how best to leverage and work with the AWS global infrastructure and AWS core foundational services. Sections 3 and 4 focus on how to optimize the architecture on AWS incorporating traditional nonfunctional requirements and how best to apply them to AWS to improve ROI. (See Figure 1.) Moving up the stack, the paper investigates alternative methods for automating the delivery of AWS using the AWS native API services and managing the infrastructure as code using the AWS cloud native system management services. Next, taking a holistic approach to AWS, the paper explores the various AWS cloud-native architectural patterns and describes the benefits a business might expect by adopting those patterns. Finally, the document will combine all these methods together, aligning to the different migration approaches useful for preparing an entire application portfolio for migration to AWS.

![Figure 1: AWS Reference Architecture for increased ROI](image-url)
3. **OPTIMIZING THE CORE INFRASTRUCTURE SERVICES**

Large legacy migration scenarios, where an organization is looking to scale its migration quickly to meet a business case, often result in the majority of applications being re-hosted on AWS. By optimizing the core infrastructure services in AWS for these migrations, a business can see a significant acceleration in ROI. Some of our AABG customers have experienced up to 50 percent reduction in hosting costs, even for applications like SAP. This section focuses on the key foundational services that companies should consider optimizing.

### 3.1 OPTIMIZING FOR NETWORK AND COMPUTE PROCESSING

Significant cost savings can be achieved with AWS network design and AWS foundational services. Companies should consider the following:

- **Region:** Unless faced with data sovereignty or performance constraints, consider regions not just based on proximity to the end user but also on cost, because regional costs can vary by as much as 45 percent. (See Figure 2.) The figure shows a summary of AWS monthly cost variations across regions. This data was calculated from a data center compute and storage bill of materials extracted from an Accenture client and cost-out for re-hosting on the AWS cloud as part of a business case.

![Figure 2: Regional pricing based on a client’s bill of materials for a data center migration](image)

- **Amazon Virtual Private Cloud (VPC) Peering:** Data transfer charges are sometimes overlooked. Leveraging VPC peering between VPCs will limit the amount of data traversing in and out of the VPC and therefore can help reduce costs.
- **AWS backbone:** The AWS backbone is a high-speed, low-cost carriage which, when utilized efficiently, can significantly lower the cost to carrier services compared to traditional telecommunication services.
- **Instance type:** Instance types and series can vary by as much as 79 percent in costs. Therefore, selecting the right instance type for the workload is critical to optimizing costs within AWS. (See Figure 3.)

![Figure 3: Instance type pricing variation](image)
• **Right size:** Over-provisioning resources in the name of performance can incur additional costs. Always provision to the lowest unit of capacity rather than over-provision. On AWS it is easier to upgrade an instance size and type.

• **Amazon Elastic Block Store (EBS) volumes:** For Amazon EBS-backed volumes AABG recommends the use of General Purpose SSD (gp2) for most production and development workloads. However, considering less-expensive Amazon EBS types for less-critical workloads is an option worth exploring. There are some exceptions such as, for example, high-performing databases like SAP HANA that require extreme high throughput. In this case, companies should consider provisioned iops (io1).

• **On demand:** Only paying for what you use presents a unique value proposition. Take full advantage of Amazon Elastic Compute Cloud (EC2) On Demand by turning workloads off when not in use. Always factor in a percentage of time a workload will not be running, especially in testing environments to help improve ROI.

• **Reserved instances:** For production workloads that are predictable, consider Amazon Elastic Compute Cloud (EC2) reserved instances to achieve up to 75 percent reductions over the term compared to on-demand pricing.

• **Spot:** To fully optimize compute costs, look at spot instances for testing and development, as well as batch and high-performing compute services, where companies can achieve reductions of up to 90 percent off standard, fully-utilized on-demand instances.

### 3.2 OPTIMIZING FOR DATA STORE AND PROCESSING

AWS has numerous services for most data processing and data storage types. A recommended approach for storage and processing is to use the most appropriate configuration of the store for the workload and data type. By selecting the most appropriate store AABG has seen customers achieve up to a 56 percent reduction in the cost to store data, as well as improved performance.11 Table 1 outlines considerations when configuring data store types in AWS for cost optimization.

<table>
<thead>
<tr>
<th>CONSIDERATION</th>
<th>CHOICE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object-based storage</td>
<td>• Standard</td>
<td>Amazon Simple Storage Service (S3) offers a range of storage classes designed for different use cases. These include Amazon S3 Standard for general-purpose storage of frequently accessed data, Amazon S3 Standard Infrequent Access, for long-lived, but less frequently accessed data, and Amazon Glacier for long-term archives. All these options differ in price and so it is important to store the right data in the right class. In addition, Amazon S3 offers configurable lifecycle policies for managing data across the various options, which allows for significant cost savings if implemented correctly.</td>
</tr>
<tr>
<td></td>
<td>• Standard Infrequently Accessed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Reduced Redundancy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Glacier</td>
<td></td>
</tr>
<tr>
<td>Scaling online transactional processing</td>
<td>• Read Replicas</td>
<td>One of the most limiting factors impacting the adoption of cloud-native applications is the challenge of optimizing a relational database which has limited ability to horizontally scale. There are techniques within AWS that can address the scaling challenges facing the database without scaling up the compute that may offer potentially more cost-effective mechanisms.</td>
</tr>
<tr>
<td></td>
<td>• ElastiCache</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Buffer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Sharding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Amazon Aurora Multi-Master</td>
<td></td>
</tr>
<tr>
<td>Non-relational database</td>
<td>• Amazon DynamoDB</td>
<td>Amazon DynamoDB charges a flat, hourly rate based on how much capacity is provisioned, which is based on the number of write and read capacity units consumed. Knowing how AWS charges for Amazon DynamoDB will help companies understand how to optimize for cost. Capacity planning is also an important aspect of ensuring cost efficiency for DynamoDB.</td>
</tr>
<tr>
<td>Online analytical processing</td>
<td>• Amazon Redshift</td>
<td>Amazon Redshift has different storage classes that can be consumed depending on cost-versus-performance requirements. Dense Storage (DS) node types enable large data warehouses using hard disk drives for a low price point. By comparison, Dense Compute (DC) node types enable high-performance data warehouses using solid-state disks.</td>
</tr>
</tbody>
</table>

Table 1: Data storage and database options within AWS
4. DYNAMIC SECURITY AND COMPLIANCE

Often one of the most effective ways to reduce IT and security costs is by migrating to AWS and adopting cloud-native security services. In some cases, AABG has seen a 96 percent reduction in the procurement of security devices and software. In addition, by adopting cloud-native security services and practices, customers can achieve a net improvement in cost avoidance, workforce productivity and customer retention.

4.1 ACCESS MANAGEMENT AND SECURITY OPERATIONS

AWS provides a cost-effective mechanism for managing access controls that is significantly more cost effective than the products used in a traditional data center. For details on how to manage access controls please refer to the Accenture Security and compliance paper for APRA as a guideline.

The Accenture Cybersecurity Engine (ACE) is a single, integrated suite of top-tier security solutions which is sophisticated enough to connect, coordinate and accelerate AWS security operations efforts. Because ACE is pre-configured and cloud-based, you can plug your security operations center (SOC) directly into ACE, which includes the processes and tools needed to take your security operations center to a higher level of performance.

4.2 FLOW CONTROLS

There are many aspects to providing flow controls in AWS, starting with an account and virtual private cloud (VPC) through to security services like the AWS Web Application Firewall and AWS Security Groups. When configured, together they provide the same (but arguably more secure) service as traditional hardware firewalls. (See Figure 4.)

In AWS, VPC flow logs can capture information about IP traffic going to and from each network interface. These logs can be used to troubleshoot connectivity and security issues as well as for anomaly and traffic analysis to increase the overall effectiveness of security operations and improve cost avoidance.

4.3 DATA PROTECTION

AWS provides several mechanisms for supporting and managing data encryption and keys depending on requirements, all of which provide improvements in automation that can improve the overall bottom line. For details on how to implement and manage data protection please refer to the Accenture Security and compliance paper for APRA.

4.4 AUTOMATED COMPLIANCE

By moving to AWS, companies no longer have to manage the physical infrastructure to ensure regulatory compliance, freeing up time for the operations team to focus on other value-added work. Equally important, moving to AWS allows environments to be standardized using approved templates. Managing the infrastructure as code, the environments can be validated using scripts to ensure that security best practices are adhered to, utilizing DevOps strategies such as zero touch deployment. AWS also supports compliance rules in AWS Config that can be automatically verified.

As a result, the compliance team can validate legal and security requirements when the system is changed, rather than relying on a periodic system review. In addition, compliance test automation can be pushed into the software development process, preventing policy violations before they are deployed into production. Capital One, for instance, has developed a rules engine to define and programmatically enforce policies on its cloud estate.
5. **CLOUD MANAGEMENT**

Operating any complex environment, especially when multiple services are deployed by many users and business units, requires additional care. Monitoring, tracking and identifying these services and their owners is not only critical from a cost perspective but is key to improving productivity. Therefore, the focus of this next topic is on lowering the cost to serve in a more effective way by focusing on agility and therefore improving productivity through the delivery of applications on AWS.

5.1 **MEASURE, MONITOR AND IMPROVE**

Monitoring is not new, but is often complex and expensive, requiring agents to be installed and managed. By contrast, Amazon CloudWatch is a cost-effective, non-intrusive and native monitoring service that provides a reliable, scalable and flexible way to monitor resource and applications on AWS. Using Amazon CloudWatch offers an alternative to traditional infrastructure management tools that are often not fit for purpose within AWS.

Another consideration for a good monitoring strategy is the use of tagging. Tags enable the ability to categorize resources by purpose, owner, environment or other criteria. A good tagging strategy is critical for cost reporting/chargeback to actual owners, giving a detailed view into cloud usage at a granular level. Tagging also provides a way for automation to shutdown instances that are not in use, or are not of an approved type, reducing technical debt and ensuring that service owners are promptly notified—e.g. via automated Amazon Simple Notification Service (SNS).

5.2 **CLOUD OPERATIONS MANAGEMENT**

One of the many ways cloud operations can help improve productivity is through the rapid delivery of environments to which developers deploy their application code. The ability to automate the delivery of infrastructure as code is one of the positively compounding differentiators of AWS, allowing the provisioning of infrastructure to developers faster than ever before. (See Figure 5.) Within AWS, every service is driven by an API. A good practice is to automate the deployment of the environments, defining infrastructure as code using services such as AWS CloudFormation. The use of CloudFormation templates allows infrastructure to be stored in source control alongside application code and other configurations.

![Figure 5: AWS Infrastructure as code](image)

5.3 **AUTOMATE DEPLOYMENT**

To further increase productivity of developers and deliver software to market faster, AWS tooling is often used to automate deployment. This requires that code deployments be automated, reliable and repeatable. In traditional data centers, deployments are often manual and sometimes error prone. Because of this, they are often deployed infrequently, leading to large deviations in the AWS environments. The greater the deviation, the greater the risk of unforeseen consequences associated with change as well as the increased difficulty in fixing environmental-related issues. Having deployments automated therefore reduces the friction associated with AWS deployment environments while ensuring that they can be performed frequently and easily. Automating the deployment of the AWS environment will increase productivity and also enable faster time to market. (See Figure 6.)

![Figure 6: AWS continuous delivery pipelines](image)
5.4 LOWER THE RISK OF DEPLOYMENTS

The ability to leverage unlimited, agile workload deployments fundamentally changes the way environments are managed and thought about when deploying software to production. Making changes to production is one of the largest risk areas for many organizations. The goal for most cloud-native architectures is, through the use of automation, to deploy straight to production. This section explores some deployment patterns that have emerged within the AWS community that can offset some of the concerns and reduce the risk of failures to production deployments. These patterns can therefore reduce the labor effort to return an expected ROI through improved productivity of the workforce involved in deployments as well as improved productivity of the operational teams required to fix them. (See Figure 7.)

Deployment patterns include:

- **Canary deployments:** The practice of directing a small number of customers to the new version and deeply scrutinizing any behavior changes or errors that are generated. If the deployment is successful, the company can continue deploying to the general population.

- **Blue/green deployments:** Alternating deployments across two identical application stacks running in parallel and sending traffic to the new version and fail back to the old version if needed.

- **Feature toggles:** Configuration options on an application that allow the deployment of software with specific features turned off. These can then be turned back on to a select user base to see the effect. If the deployment has problems, simply turn the feature back off.
6. RESILIENCE AND BUSINESS CONTINUITY

Designing applications for higher levels of availability typically comes with increased costs, so companies need to identify their true objectives for availability and recovery point objectives before embarking on an architecture pattern on AWS. When considering resilience and business continuity in a traditional data center model, the most significant components related to cost are the needs for additional infrastructure, and in some cases an entirely new facility. Furthermore, some of the most challenging and often expensive mechanisms required to support resilience are the network components. This section provides a guide on how to use the AWS regional networking constructs to leverage the equivalence of several data centers to achieve reliable and cost-effective application patterns.

6.1 SINGLE-REGION DESIGNS

The most pervasive fault isolation construct in AWS is that of the region. Regions are designed to be autonomous, with dedicated copies of services deployed in each. Regional AWS services internally use multiple Availability Zones (AZs) where each AZ contains one or more physical data centers clustered together. Each AZ is separated by a large-enough physical distance to avoid correlated failure scenarios, and each AZ has an independent physical infrastructure.

One of the defining differentiators of AWS is the fact that, despite being geographically separated, AZs are located in the same regional area, allowing for single-digit, millisecond latency between them. Cloud-native architecture patterns leverage this design to use AZs in an active/active or active/standby configuration. From an ROI point of view, this architecture is significant because it enables the ability to design for synchronous data replication across AZs without having to invest in the kind of network infrastructure found in a more traditional data center model, with minimal impact on application latency. (See Figure 8.)

Table 2 lists the Service Level Agreements for the availability of a standard, three-tier application in a single AWS region, utilizing backup tools and automated deployment tools and services described previously in section 5.3. Details of the application architecture used in Table 2 can be referenced in the AWS paper, “Reliability Pillar: AWS Well-Architected Framework.”

<table>
<thead>
<tr>
<th>AWS DEPLOYMENT</th>
<th>SERVICE LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-availability zones</td>
<td>99%</td>
</tr>
<tr>
<td>Dual-availability zones</td>
<td>99.9%</td>
</tr>
<tr>
<td>Multiple-availability zones</td>
<td>99.99%</td>
</tr>
</tbody>
</table>

Table 2: Resilience service levels for various regional deployments for a typical three-tier application
Designing applications for resilience in multiple AWS regions can increase the cost of operations, partly because AWS isolates regions to maintain independence. The decision to take this path should only be made if absolutely necessary. An example of where AABG has seen this is when a client has been concerned about the effect of seismic activity on one region. Regions provide a very strong isolation boundary and AWS by design goes to great lengths to avoid correlated failures across regions. (See Figure 9.) Therefore, using multiple regions will provide greater control over recovery time in the unlikely event of a hard dependency failure on a regional AWS service. Table 3 highlights the various implementation patterns and their typical availability.

<table>
<thead>
<tr>
<th>AWS DEPLOYMENT BETWEEN REGIONS</th>
<th>SERVICE LEVELS ACHIEVABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active/Passive</td>
<td>99.95 percent</td>
</tr>
<tr>
<td>Active/Active</td>
<td>99.999 percent</td>
</tr>
</tbody>
</table>

Table 3: Resilience service levels for cross-regional deployments for a three-tier application
7. **MONETIZING BUSINESS FLEXIBILITY**

Traditionally, a business would perform detailed capacity planning and make large upfront investments in hardware and software to accommodate demand. Now, moving to a model that automates a “just enough” capacity model from an ROI perspective ensures commoditizing for cost, and allows the ability to monetize the business flexibility by meeting customer demands.

One of the defining features of AWS is the ability to auto-scale. The AWS Auto Scaling capability allows for the automatic scale-out of Amazon Elastic Cloud Compute (EC2) instances to handle the load of an application. To achieve this goal, AWS Auto Scaling dynamically regulates compute capacity requirements based on changes in Amazon CloudWatch-monitored metrics or at a scheduled time. Figure 10 shows how different days of the week typically have different capacity requirements. Using AWS Auto Scaling, the right amount of capacity can be provisioned, leaving only minimal excess capacity.

Table 4 explores different load volatility patterns and the impact on ROI when using AWS Auto Scaling to reduce leakages, improve performance and reduce costs.

<table>
<thead>
<tr>
<th>LOAD PATTERN</th>
<th>DESCRIPTION</th>
<th>TYPICAL APPLICATION TYPE</th>
<th>POTENTIAL SAVINGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly spikes and troughs</td>
<td>This pattern describes workloads that have peak usage for 12 hours between 8:00 am to 8:00 pm in a day. For the rest of the day, the capacity is usually under-utilized.</td>
<td>• System of record</td>
<td>Since lower utilization is typically observed during the off-peak periods at night, capacity can be reduced in an automated way to save costs by up to 25% over fully utilized services.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ecommerce</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ERP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weekly fluctuations</td>
<td>This pattern is observed by workloads which have peak usage for 4-5 days a week, then under-utilization in the build-up and on the weekend.</td>
<td>• Online applications</td>
<td>These dynamic workloads are a good use case for AWS and AWS Auto Scaling, resulting in up to 30% savings over fully utilized services.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Customer record</td>
<td></td>
</tr>
<tr>
<td>Seasonal fluctuations</td>
<td>This pattern usually has seasonal peaks for 3-5 months with heavy utilization. The rest of the year is normally characterized by under-utilized capacity.</td>
<td>• Industry-specific workloads</td>
<td>Using AWS Auto Scaling systems can be scheduled to gradually increase EC2 instance capacity days before the season and gradually decrease capacity over the weeks after the peak season. This approach can deliver more than 40% savings over fully utilized services.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ticketing systems for events</td>
<td></td>
</tr>
</tbody>
</table>
7.1 EASIER TO REBUILD THAN TO REPAIR

AWS enables companies to quickly anticipate operational events, both planned and unplanned, using existing deployment pipelines to deliver consistent results when responding to an event.

AWS simplifies response to events by providing tools that supports all aspects of a workload and code deployment. These tools allow scripted responses to operational events and triggers. In AWS, it is possible to significantly improve recovery time by replacing failed components with known good versions, rather than trying to repair them. This provides the ability to quickly rebuild and recreate whole environments and applications on demand instead of spending excessive hours trying to manually repair these environments when something goes wrong. Embracing this mindset aligns to a cloud-native approach to application resilience and management which can lead to decreased technical debt and a lower total cost of ownership (TCO).

7.2 MANAGED SERVICE OFFERINGS

Having already established its core, foundational infrastructure building blocks, AWS has built an assortment of managed service offerings that offer value as either an add-on or alternative to the services the traditional self-hosted workloads rely on. These services often come integrated with the rest of the corresponding AWS services and apply the same rigor around performance, agility, scalability and security as the core infrastructure services. AWS provides a viable alternative to what might be consumed in a data center today at a fraction of the price. AWS offers new services to support customers and their business requirements. Consuming these services not only reduces the managerial overhead but also provides the business with yet another mechanism to monetize AWS.

7.3 DATA-DRIVEN INSIGHTS

The AWS API-driven model enables a company to capture as much telemetry data as possible about an environment and application behaviors. Cloud-native organizations have adopted such techniques to identify and discover variances and even weak signals within the data collected that can then be used to avoid failures. Companies that have leveraged AWS to capture all metrics and API calls have adopted data insight techniques to identify, for example, if EC2 nodes in a cluster are deviating from normal behavior patterns. If an abnormality is detected the application and environment code will then terminate the node and restore a new one, with an established, stable build, all without any human intervention. Such techniques have resulted in organizations being able to significantly reduce the time spent on un-planned outages, improving the productivity of developers and operations teams.
8. SOFTWARE DELIVERY AND PROCUREMENT

Changing how companies and public-sector organizations they think about procurement can be a bigger barrier to cloud adoption than security concerns. A significant challenge many procurement officers in the enterprise face with cloud adoption is moving from CAPEX-based to OPEX-based procurement. One issue is that procurement executives need to figure out the mechanism for annual budgeting when there is no upfront cost. Another impediment to cloud adoption is the commercial support for enterprise applications whose software has yet to embrace the distributed model of the cloud. Other issues include the ability to keep pace with the rate of change and the continual need to align the business, developers and operators with the right tools and software to meet a company’s ever-changing requirements. The AWS Marketplace is an online store where software can be bought and sold that runs on AWS On-Demand using a utility model. It offers an alternative to address some of the complexities faced when transitioning from traditional procurement models to more cloud-native approaches to software procurement.

8.1 DISTRIBUTING SOFTWARE THROUGH AWS MARKETPLACE

AWS Marketplace offers a variety of third-party, cloud-native software products that organizations can consume under a single-billing, procurement, governance and security model. With more than 4,200 software listings and 160,000 active customers consuming the software, AWS Marketplace is one of the largest and most subscribed-to software repositories in the world. Some of the benefits of leveraging the AWS Marketplace as a procurement vehicle are listed in Table 5.

<table>
<thead>
<tr>
<th>BENEFITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management overhead</td>
<td>All the heavy lifting of installing, configuring the software and updating with the latest patches is managed by the software vendor through AWS Marketplace.</td>
</tr>
<tr>
<td>Speed to deploy</td>
<td>One of the key benefits of AWS Marketplace is that the deployment process is often much quicker and smoother than traditional software installations. There is no manual installation, no lengthy configuration process, and no additional hardware requirements on the part of the user.</td>
</tr>
<tr>
<td>Pricing simplification</td>
<td>With products bought through the AWS Marketplace, payment is simple. All billing is contained within the AWS Marketplace and payment is taken using existing AWS billing details. This pricing is set by the vendor, but is clearly stated on the AWS Marketplace page, along with the usage fees for related web services such as running an instance on Amazon EC2.</td>
</tr>
<tr>
<td>Fit for purpose</td>
<td>With many of the products available on the AWS Marketplace, there are often a number of ways for companies to “try before they buy.” This option offers an agile, secure and reliable way of getting hands-on experience with the tool before committing to becoming an hourly or annual license holder.</td>
</tr>
<tr>
<td>Centralized procurement</td>
<td>Through AWS Marketplace, companies have a one-stop-shop for their software procurement, eliminating long lead times and providing the business the opportunity to experiment and try out software without significant financial commitments.</td>
</tr>
<tr>
<td>Improve latency</td>
<td>Having needed applications sitting within the AWS ecosystem reduces the integration and network latency between applications, something that is particularly relevant with a microservices architecture.</td>
</tr>
<tr>
<td>Compliance and governance</td>
<td>Companies can standardize on a secure and compliant landscape with the assurance that software vendors will provide support.</td>
</tr>
<tr>
<td>Accounting</td>
<td>The AWS Marketplace offers an OPEX model by which organizations can simplify and streamline licensing and billing by adopting a consumption-based model, avoiding software commercial lock-in.</td>
</tr>
</tbody>
</table>

Table 5: Benefits of procuring software through the AWS Marketplace
8.2 COMMERCIAL SUPPORT FOR ENTERPRISE SOFTWARE

With the adoption of cloud, software licensing complexity has increased. A key issue with licensing in a cloud environment is how software is purchased and allocated across instances of the cloud, and how the use of licenses is monitored and tracked. Organizations can mitigate some of the complexity associated with licenses by taking advantage of the commercial software of the AWS Marketplace where the vendors have reworked the licensing model to be in line with the cloud, usually removing licenses tied to things like CPU cores.

Alternatively, companies should take care when identifying software and licensing that could be migrated to AWS. The best migration plans are those that consider licensing from the very beginning. No enterprise wants to be caught mid-audit with hundreds of users violating an end-user license agreement. (See Table 6.)

### LICENSING OPTIONS

<table>
<thead>
<tr>
<th>LICENSING OPTIONS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>License included</td>
<td>On Amazon EC2, companies can choose to run instances that include the relevant license fees with their on-demand pricing.</td>
</tr>
<tr>
<td>Bring your own license</td>
<td>Companies can buy their own license (BYOL)—a model used to deploy software that has been previously licensed on physically dedicated hardware. If using BYOL, only the cost of the instance is paid, not the software running there. Note that some software vendors have specific requirements about how to utilize the terms of the license, such as procuring dedicated hosts or the number of CPU cores that comply to a license unit.</td>
</tr>
<tr>
<td>License mobility</td>
<td>Some software vendors have license mobility, a benefit available to licensing customers with eligible server applications covered by active software assurance. License mobility allows customers to move eligible software on AWS for use on Amazon EC2 instances.</td>
</tr>
</tbody>
</table>

Table 6: Commercial considerations for enterprise software hosted in AWS

8.3 SEEKING OUT OPEN-SOURCE ALTERNATIVES

From the beginning, AWS has embraced Linux and the open-source community. Organizations can readily take advantage of open-source software in the AWS Marketplace and can try out and research alternatives to commercial software without having to install and configure, safe in the knowledge that AWS has securely reviewed the application and that a structured governance framework is in place. Organizations committed to open source can often move to AWS with reduced risk. See Table 7 for some of the benefits of an open-source strategy.

### LICENSING OPTIONS

<table>
<thead>
<tr>
<th>LICENSING OPTIONS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affordability</td>
<td>Open-source licenses are free. Despite having some overhead to maintain, install and support, the ROI of open source is usually considerably better than commercial alternatives.</td>
</tr>
<tr>
<td>Flexibility</td>
<td>When using open-source applications, organizations have the freedom to choose what solutions are best for a specific use case, without needing to lock into contracts.</td>
</tr>
<tr>
<td>Sustainability</td>
<td>Most open-source software is updated on a regular basis by the community that supports it. This updating provides the ability to enhance products versus replacing them and, as a consequence, ultimately falling behind.</td>
</tr>
<tr>
<td>Secure</td>
<td>Whereas proprietary software is maintained by a small team at a single company, open-source projects are often actively implemented and maintained by a large community of specialists who can address security vulnerabilities much faster.</td>
</tr>
<tr>
<td>LICENSING OPTIONS</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Reliability</td>
<td>Because open-source technology is fed by a community without the burden of deadlines for delivering new features, the solutions implemented are often more reliable.</td>
</tr>
</tbody>
</table>

Table 7: Some of the benefits of adopting open-source software

9. **RE-ARCHITECTING APPLICATIONS FOR AGILITY TO ADD VALUE**

Re-architecting requires a detailed understanding of the services and benefits AWS has to offer and an in-depth knowledge of the application. This knowledge includes the application’s functionality, data, non-functional requirements and usage patterns—all toward providing the understanding and ability to refactor and maximize AWS. Cloud-native applications that take advantage of cloud-native features requires applications to follow a distributed and scalable service-oriented framework with built-in redundancy to handle failures. The sections below discuss the cloud-native architecture patterns which, when combined and built using AWS services, can fully realize the value of AWS and improve overall ROI.

9.1 **RESILIENT ARCHITECTURE**

One of the bedrock principles for service design in AWS is the avoidance of single points of failure in underlying physical infrastructure (discussed in Section 7). This avoidance results in building software and systems that use multiple Availability Zones and are resilient to failure within any single zone. Similarly, resilient architecture patterns are built to be resilient to the failure of a single compute node, single storage volume, or single instance of a database. Figure 11 shows multiple available zones and system components deployed in each zone, distributing the load so that, in the event of faults, outages are avoided.

![Figure 11: Resilient architecture pattern for multiple Availability Zones (AZs)](image)

This architecture pattern will increase resilience, lowering the cost to meet the business objectives and help the operations team reclaim time and energy that can then be spent on servicing the developers and the business more efficiently.

9.2 **DISTRIBUTED SYSTEMS**

Many enterprise services are distributed systems that rely on messaging interfaces for interoperability with other components of the system. Microservices architectures in particular require complex interconnections between components of the system. That is why, in cloud-native architectures, it is vital to have a robust API architecture. A good API architecture can support the reliable, secure and fast transfer of data across the network and to the distributed components that make up the cloud-native applications. AWS has an API service, Amazon API Gateway, which will manage and distribute a company’s API strategy to service the cloud-native distributed systems hosted in AWS. (See Figure 12.)
9.3 MICROSERVICES

A microservices architecture is a design approach for building a single application as a set of small services. Each service runs in its own process and communicates with other services through a well-defined interface using a lightweight mechanism—typically an HTTP-based API. Microservices are built around business capabilities. In general, each service is dedicated to a single function. Many different frameworks or programming languages can be used to write microservices. Microservices can be deployed independently, as a single service, or as a group of services. (See Figure 13.)

Microservices allow systems to hyper-scale. They provide increased reliability over traditional distributed system architectures given the decoupling of functional application dependencies within the application code. The decoupling of microservices reduces the risks related to changes in the system. From an ROI standpoint, decoupling is one of the key architecture designs that allows organizations adopting cloud-native applications to deliver new services to market faster than their competitors. Microservices architectures provide significantly reduced time and technical overhead managing unplanned work. This reduced technical debt allows developers to spend more time on developing products, increasing developer productivity.

AWS provides many services that can help design and build out an application to be more loosely coupled and adopt a microservices architecture approach. Table 8 details some of the key services.

<table>
<thead>
<tr>
<th>AWS SERVICE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Lambda</td>
<td>AWS Lambda lets code run without provisioning or managing servers. Just upload code, and Lambda manages what is required to run and scale the code with high availability, making it easier to create functions serviced as microservices.</td>
</tr>
<tr>
<td>Amazon ECS</td>
<td>A highly scalable, high-performance container management service that supports docker containers and allows microservices to more easily run on a managed cluster of Amazon EC2 instances.</td>
</tr>
<tr>
<td>AWS SERVICE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Amazon API Gateway</td>
<td>Amazon API Gateway allows the processing of hundreds of thousands of concurrent API calls. It handles traffic management, authorization and access control, monitoring, and API version management for microservices interactions.</td>
</tr>
<tr>
<td>Amazon Aurora</td>
<td>A relational database engine that combines the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases.</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>A fully managed, fast and flexible NoSQL database service appropriate for all microservices applications that need consistent, single-digit, millisecond latency at scale.</td>
</tr>
<tr>
<td>Amazon Simple Queue Service</td>
<td>A fully managed message queuing service that makes it easy to decouple and scale microservices, distributed systems and serverless applications.</td>
</tr>
</tbody>
</table>

Table 8: Some of the key AWS services to help facilitate the design of microservices

9.4 REACTIVE SYSTEM ARCHITECTURE

Building on the microservices architecture, systems built as reactive systems are more flexible, loosely-coupled and scalable. This makes them easier to develop and more amenable to change. Reactive systems are significantly more tolerant of failure. When failures do occur, they are more contained, rather than cascaded throughout the environment. Reactive systems are:

1. **Responsive**: Problems within the system are detected faster and dealt with more effectively. Responsive systems focus on providing rapid and consistent response times.

2. **Resilient**: The system stays responsive in the event of failure. Isolation and bounded context applied to microservices helps to ensure that parts of the system can fail and recover without compromising the whole system.

3. **Elastic**: Reactive systems can react to variance in the workload by increasing or decreasing the resources allocated to the service on demand.

4. **Message-driven**: Reactive systems rely on asynchronous messaging to establish the means to delegate failures as messages so that the system can deal with component failures with no impact to the system.

9.5 SERVERLESS ARCHITECTURES

Serverless architectures provide the ability to build and run applications and services without needing to manage the infrastructure, including scaling, security, patching and resilience. With serverless applications, organizations do not need to provision, scale or manage instances, allowing developers to focus on their application so they can accommodate nearly any type of application or backend service.

Building serverless applications means that developers can focus on the core product instead of worrying about managing and operating servers or runtimes. This reduced overhead lets developers reclaim time and energy, which can now be spent on developing products, increasing developer productivity and lowering the operational effort needed to support them.
Figure 14: AWS Lambda used within an AWS serverless architecture

AWS Lambda executes an organization’s code only when needed and scales automatically, from a few requests per day to thousands per second. (See Figure 14.) Organizations pay only for the compute time consumed. There is no charge when code is not running. This AWS service, used in a microservices architecture, can provide high returns on investment when working on cloud-native design patterns. Lambda functions only occur when the actual code is being processed, therefore, depending on how often the code is executed, can result in significant cost reduction, as high as 95 percent.27

10. BRINGING IT ALL TOGETHER: APPLICATION AND SERVICES RATIONALIZATION

The complexity of moving applications to AWS varies considerably depending on the architecture and—for non-custom-developed applications—on existing commercial arrangements. This document has discussed in detail how to leverage the AWS foundation services and core services to help optimize the use of AWS. We have also discussed various application architectures that can be leveraged to move towards a cloud-native services to help improve ROI. This section will outline how to apply these concepts to some of the migration patterns and application rationalization techniques when considering a holistic, cloud-first strategy for AWS.

10.1 APPLICATION RATIONALIZATION STRATEGY

To gain the full value of AWS, companies need to leverage the entire breadth of the services and capabilities. Looking at the applications portfolio and workload patterns provides important keys to unlocking how to better monetize AWS. (See Figure 15.) In a well-known blog by Stephen Orban of AWS, he discusses strategies for migrating applications to the cloud. Here, we will focus on four strategies in particular:

Figure 15: Process flow for application migration path
• **Re-hosting**, a simple “lift and shift” of an application in its current form to AWS while optimizing the use of AWS foundational services

• **Re-platforming**, which does a partial re-architecture to leverage some of the AWS services on offer

• **Re-purchasing**, seeking out commercial or open-source alternatives that will align more to cloud-native adoption and avoid software lock-in.

• **Re-architecting**, taking advantage of cloud-native capabilities

Which migration path to take depends on several factors that acknowledge both technical and business requirements. No matter the application migration path to AWS, the business will incur some upfront investment in migrating. (See Figure 16.) Conversely, staying within the data center will at some point trigger a hardware lifecycle purchase as well as application license renewals if targeting commercial software. Deciding to do nothing can often be more costly over time.

![Figure 16](image)

Determining the break-even-point is key to determining which application migration decision path to plan for. Each application breakeven-point will be unique to organizational requirements, the infrastructure and the application. These factors will, in turn, determine the optimal migration approach and the most effective point in time to execute the migration. Determining the migration approach will dictate the initial upfront investment and ongoing costs for AWS. Factoring the point in time that the application will require a hardware purchase to continue running in the existing data center will also play a factor in the ultimate decision on when and how to migrate. All these factors need to be considered and will conclude the break-even-point of a given application.

Worthy of note, however, is that the above is just based on cost takeout and does not include the returns on improved productivity and time to market that we have discussed as a theme throughout this document.

### 10.2 CONCLUSION

Something as simple as selecting and leveraging the right AWS foundational services can have profound impacts on an organization’s IROI when rehosting applications. Even greater value is realized when companies start changing the way they think about application design and the various methods of migrating an application portfolio to AWS. A holistic approach to leveraging AWS delivery and application services helps define how to re-platform applications in a migration journey. Re-platforming applications can not only lower an organization’s bottom line, but can also start the journey towards improving the top line in terms of agility, customer retention through cost avoidance, and a more effective workforce.

The power of AWS can be improved when companies take a significant step towards refactoring and repurchasing applications using cloud-native architectures and the full stack of available AWS services. Organizations looking to gain all the benefits that AWS is providing today can adopt cloud-native architectures, delivering significant improvements to the top line and accessing new lines of business to start differentiating the organization in a competitive market.
10.3 ABOUT ACCENTURE

Accenture is a leading global professional services company, providing a broad range of services and solutions in strategy, consulting, digital, technology and operations. Combining unmatched experience and specialized skills across more than 40 industries and all business functions—underpinned by the world’s largest delivery network—Accenture works at the intersection of business and technology to help clients improve their performance and create sustainable value for their stakeholders. With approximately 452,000 people serving clients in more than 120 countries, Accenture drives innovation to improve the way the world works and lives. Visit us at www.accenture.com.

This document makes descriptive reference to trademarks that may be owned by others. The use of such trademarks herein is not an assertion of ownership of such trademarks by Accenture and is not intended to represent or imply the existence of an association between Accenture and the lawful owners of such trademarks.

This document is produced by consultants at Accenture as general guidance. It is not intended to provide specific advice on your circumstances. If you require advice or further details on any matters referred to, please contact your Accenture representative.

---

1 Accenture research based on client engagements
2 Accenture research based on client engagements
3 Accenture research based on client engagements
4 Accenture research based on client engagements
5 Accenture research based on client engagements
6 https://aws.amazon.com/architecture/well-architected/
7 Accenture research based on client engagements
8 For more, see https://aws.amazon.com/ebs/details/
9 https://aws.amazon.com/ec2/pricing/reserved-instances/
10 https://aws.amazon.com/ec2/spot/
11 Accenture research based on client engagements
12 For more details see https://aws.amazon.com/s3/
13 https://aws.amazon.com/dynamodb/
14 https://aws.amazon.com/redshift/
15 Accenture research based on client engagements
26 Accenture research based on client engagements