Migrating Magento® eCommerce Platform to AWS

April 2020

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

Adopting AWS for the Magento e-commerce software platform presents many benefits such as increased business agility, flexibility, and reduced costs. This whitepaper outlines the benefits of cloud hosting for Magento and considerations for migrating an on-premises Magento installation to AWS. Also, it provides guidance for an organization that plans to increase their cloud footprint. The content targets technical leaders and business leaders responsible for deploying and managing Magento on AWS.

Introduction

Creating a new or migrating an existing Magento setup to Amazon Web Services (AWS) Cloud presents an opportunity to transform your organization by lowering costs, increasing agility, and deliver reliably and globally. This whitepaper presents a cloud migration strategy and considerations when migrating Magento to AWS.

This whitepaper provides general guidance for cloud migration with specific guidance related to migrating a Magento installation to cloud. In addition, the paper provides guidance to enable an organization that wants to expand their use of cloud. The first section of the whitepaper describes the reasons to migrate to the cloud and the common challenges that organizations face when migrating to the cloud. Then, the migration process and the migration strategies that organizations can choose from as well as deployment options are discussed. Lastly, the whitepaper concludes by discussing security and compliance, architectural components, connectivity and a strategy you can employ for migration.

Drivers for Migrating to Cloud

The drivers behind starting a new Magento setup or moving an existing on-premises Magento setup to the cloud are numerous but the most common strategic drivers include: reducing capital expenditure, decreasing ongoing cost, improving scalability and elasticity, improving time-to-market, and attaining improvements in security and compliance. In addition, situational and business drivers also influence the move to the cloud.

DevOps

Supporting your organization's DevOps strategy by migrating to the cloud may be a primary driver for migrating to cloud or may be an unanticipated benefit. In either case, migrating to cloud provides a technical foundation to supporting your organization's DevOps strategy by way of the same capabilities expected of cloud and as defined by NIST[1]: on demand and self-service, broad network access, pooled resources, rapid elasticity, all delivered as a metered service providing you the ability to control how your organization consumes it.

Each of these capabilities directly maps to demands placed on a technology organization, regardless of your organization's adoption of DevOps, Site Reliable Engineering (SRE), etc., but of most import is the ability to programmatically define

infrastructure and configuration (infrastructure as code [IaC]) and using that ability to dynamically create/tear down environments as part of a well implemented software development life cycle (SDLC) process.

In addition to providing a supporting technology platform for the enablement of DevOps processes on AWS around your Magento environment, AWS provides a collection of services that can provide (in the absence of) or augment your existing software configuration management (SCM) solutions, including AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, and AWS CodeDeploy, which provides for a managed source control, build, continuous integration/continuous deployment (CI/CD) and deployment services.

Data Center Consolidation

Data center consolidation is a key requirement driver that may warrant the need to move to the cloud. For example, an organization's current data center can no longer support the business need for growth in terms of current space, power and cooling. Also, an organization's current data center may have too many single points of failure and carry inherent risks of outages.

Mergers and Acquisitions

Mergers and acquisitions are a situational driver. Mergers or acquisition prompt an organization to separate and/or consolidate application setup due to a merger or acquisition. Also, an organization may face the sale of a building, rental fees, or increases in co-location costs that may result in similar needs to consolidate or separate application setup, leading to the need to move to the cloud.

Digital Transformation

Digital transformation is more than simply digitizing data and involves the transformation of business and organizational activities, processes, competencies to accelerate deliverables that differentiate an organizations core business. The need for digital transformation in organizations have resulted in the evolution of organization's IT department to become more agile and innovative to adapt to the changing needs of an organization.

AWS Cloud infrastructure setup frees an organization's IT department from the heavy lifting of racking, stacking, and powering servers to focus on the organization's own customers. Concentrating on the projects that differentiate an organization's core

business, rather than the infrastructure, substantially improves products, services, delivery, and ultimately the ability to compete.

Benefits of Cloud

Organizations considering a transition to the cloud are often driven by their need to become more agile and innovative. The traditional capital expenditure (Capex) funding model makes it difficult to quickly test new ideas. The AWS Cloud model gives you the agility to quickly spin up new instances on AWS, and the ability to try out new services without investing in large upfront, sunk costs (costs that have already been incurred and can't be recovered). AWS helps lower customer costs through its pay for what you use pricing model.

Operational Improvements

The value proposition of migration of Magento to AWS is further enhanced by the availability of several services that provide for operational insight and agility.

Operational insight into the platform from not only a technical perspective (e.g. requests per hour) but also a business operational perspective (e.g. orders per hour, etc.), particularly when the two sets of data can be married, provides a near-real-time look into campaign performance, platform operations costs, and a near infinite number of other indicators.

This data provides as basis for, and support of, change as does the agility afforded by AWS, allowing for a content and functional deployment pipeline, a/b testing and feedback loops, allowing for continuous improvement, measurement and pivoting when it comes to the user experience of the Magento store.

Technology Improvements

The momentum of a migration presents an opportune time to look at technical improvements. Enabled with AWS, these technical improvements can be implemented in a commitment-free manner, evaluated and codified.

A typical Magento installation, on premise, may leverage as little as one server or numerous servers, based on scale and architecture, but in addition to servers, third party services may be leveraged as well (content delivery networks, indexing services, etc.).

The breadth and depth of the AWS service offerings, as well as the billing constructs available on AWS serve to eliminate the baseline quantity of servers to be cared for and

fed as well as provide a means to consolidate vendors, achieve larger quantities of scale and have predictable baseline and burst cost models as it relates to operating the infrastructure. This is exclusive of the potential gain in efficiency around management of the platform, furthering lowering operating costs.

Cloud Adoption Challenges

Cloud has become a key pillar of most enterprises' digital transformation strategies. Organizations are both migrating new processes to the cloud and augmenting their existing cloud operations to take advantage of new and evolving services. However, these are complex initiatives that many organizations stumble through because of the following constraints and concerns.

Cost/Budget Constraints

A core reason why organizations adopt a cloud IT infrastructure is to save money. However, there are budget and cost constraints about the moving and running operations in the cloud that often pose challenges for organizations. AWS offers cost estimation and budgeting tools, such as AWS Cost Explorer, AWS Cost and Usage Reports, and AWS Budgets, that can guide organizations and alleviate these concerns.

Security Concerns

Organizations are accustomed to having full ownership from the physical building that house the servers to the software on the servers. This ownership made people comfortable that the data was secure. Ownership and the geographic placement of data have become major topics for cybersecurity and cloud policy initiatives around the globe.

As technology has evolved, however, most threats are exploited remotely. The physical location of data has little to no impact on threats propagated over the Internet. Organizations often have the misconception that cloud environment is less secure. In addition, organizations IT departments are well versed with security of on-premises infrastructure but often struggle with implementing security in the cloud due to lack of knowledge or skills.

Responding to Business Requirements

Organizations struggle to identify the different end customers such as shoppers, content creators, customer service agents, merchandizers etc. in a typical e-commerce

application setup. This becomes a challenge in a cloud migration initiative to respond and work backwards from the requirements for each customer type.

Managing Legacy Infrastructure

Organizations may not realize the cost savings immediately from moving Magento application to the cloud because in some cases the on-premises infrastructure cannot be decommissioned as other services/applications are still running using that infrastructure. This results in expanding the scope for organizations to manage existing legacy infrastructure along with the cloud infrastructure.

Competing Projects, Staffing Concerns & Skills Shortage

Organizations struggle with cloud related skills shortage in existing staff and staffing concerns to hire cloud trained staff. In addition, competing priorities and projects between multiple cloud initiatives at an organization lead to the IT staff split between projects. This ironically leads to slow and costly processes to implement and optimize systems meant to deliver speed and agility. AWS Cloud and managed service providers, such as CloudHesive, offer support services, which have become much more proactive and strategically focused in response to market demand for increased levels of responsiveness, access to tools, and strategic guidance.

Vendor Management

Organizations often lack resources to manage vendors and see cloud service providers an addition to that list. In addition, getting the right documentation, contract management and compliance reports are other struggle areas that add to the complexity. AWS, however, simplifies these challenges with its self-service approach and availability of compliance reports online.

Five-Phase Migration Process

When considering the migration of a Magento workload to the cloud, it is typically just one part of an overall migration plan. For example, a company embarking on a modernization journey to move from a data center or co-location facility to AWS must develop a detailed migration plan that considers the sequence and strategy for moving applications and platforms with the least impact on ongoing business operations.

This section covers the five-phase migration process in the context of migrating a Magento workload to the cloud.

Phase 1: Migration Preparation and Business Planning

In this phase, you gain a complete understanding of the benefits of migrating to the cloud as well as establishing your business objectives. A business case for the migration is developed and the constraints of existing architectures and systems are considered. This is also where AWS partners who specialize in cloud migrations and Magento deployments can be engaged to help develop a migration plan.

Phase 2: Portfolio Discovery and Planning

Next, you inspect your entire IT portfolio, understanding any dependencies that exist between workloads, and consider the migration strategies to meet your business objectives.

For Magento, this phase details how Magento integrates with other workloads, both internal and external to the business.

Phase 3 & 4: Designing, Migrating, and Validating Applications

This is where the migration strategy for each application is designed, performed, and validated. There are six common application migration strategies which are discussed in more detail in the next section.

Phase 5: Modern Operating Model

Finally, this phase is where you iterate on your new foundation, retiring old systems, and continuing to improve and move toward a modern operating environment.

Migration Strategies

This section provides the six common migration strategies for moving applications and systems to the cloud and then describes which of those strategies apply to Magento workloads. It's useful to understand these broader strategies since Magento is typically just one component of a portfolio of applications and therefore part of an overall migration plan.

Six Common Strategies: "Six Rs"

The six approaches described below are common migration strategies and build upon "The 5 Rs" outlined by Gartner in 2011[2]. Choosing the right Magento migration strategy

depends upon the business drivers for cloud adoption, as well as time considerations, business and financial constraints, and resource requirements.

Re-host

Rehosting, or “lift and shift”, is typically used when an organization is looking to quickly migrate applications to the cloud to meet a business case. Applications are moved as-is to the cloud without making any changes to the application or its dependencies. Although this strategy does not immediately bring the full benefits of the cloud, it allows for a swift migration and cost savings from hosting in the cloud.

Re-platform

Also referred to as “lift, tinker, and shift”, re-platforming involves taking an existing application, migrating it to the cloud, and replacing specific application dependencies with fully managed alternatives available in the cloud. For example, rather than directly hosting a relational database on EC2 instances, the database for many applications can be easily replaced by Amazon Relational Database Service (Amazon RDS). The benefit to this strategy is that the operating responsibility of undifferentiated components can be offloaded to AWS without requiring significant changes to the core application.

Re-purchase

Re-purchase strategy involves moving from perpetual licenses to a software-as-a-service (Saas) model. In context of Magento, a Re-purchase strategy would involve moving from a traditional on-premise Magento license (often referred to as M1) to Magento Commerce Cloud (a hosted software-as-as service).

Re-factor / Re-architect

Re-factor or Re-architect strategy gives the most opportunity to optimize and re-skin or re-imagine the application architecture from ground up. This presents an opportunity to deliver features using cloud native technologies. This strategy is often driven by strong business need to add features, scale, or performance that would otherwise be difficult to achieve in the application’s existing environment.

Retire

In context of Magento, this strategy involves completing a discovery of the existing environment and removing applications/features that are no longer needed. For

example, hardware monitoring may not be required if you are planning to move to a managed Magento commerce solution.

Retain

This is also referred to as a “re-visit” strategy or do nothing. This strategy involves revisiting the existing Magento application at a later point in time because migrating to cloud may not align with current business needs.

Magento Versions

Magento is available in two versions: 1) Open Source (formerly known as community edition) and 2) Magento Commerce. (See the feature comparison on the Adobe Magento site for version differences.)

Magento Open Source is available only for self-hosting whereas Magento Commerce is available for self-hosting or as part of Magento Hosted Cloud (a.k.a. Magento Commerce Cloud).

Reference Architecture

The following figure shows the reference architecture for deploying either version of Magento on AWS. For reference architecture details, see Hosting Magento® eCommerce Software on AWS.

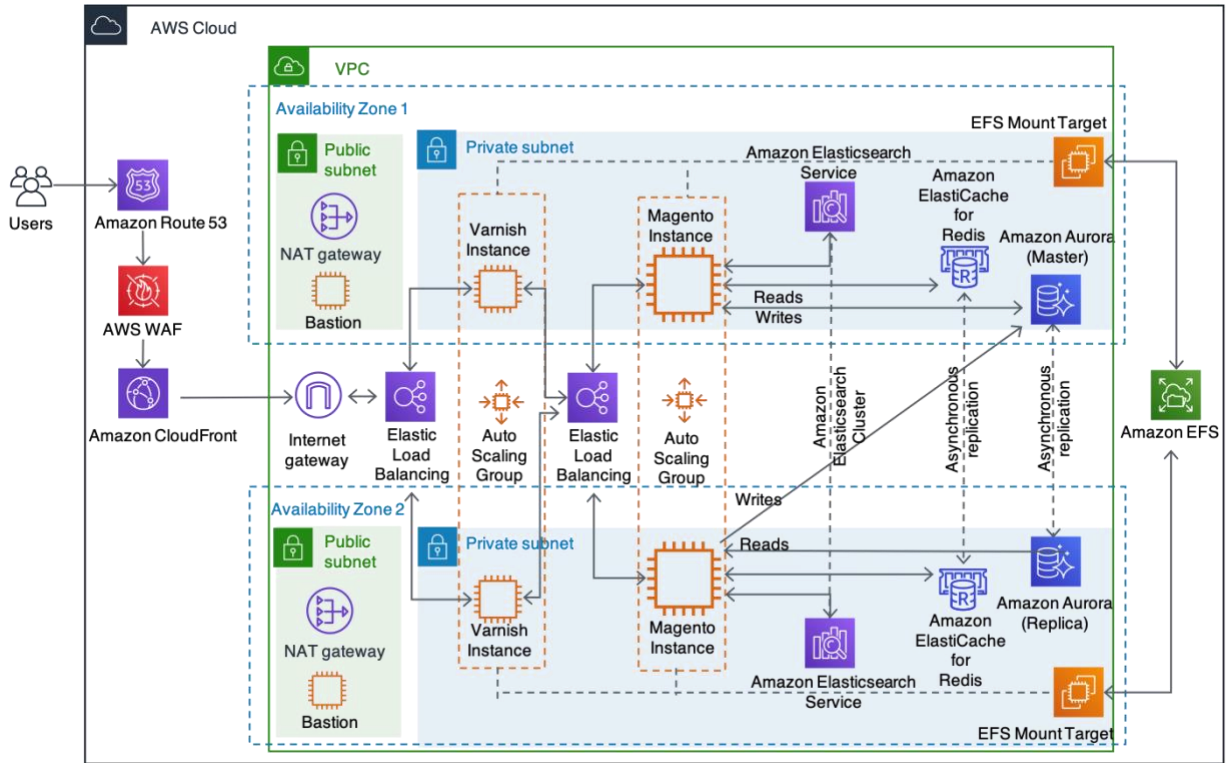

Figure 1 – Reference architecture for Magento Commerce and Magento Open Source

Deployment Options

There are several deployment options available for running Magento (both Open Source and Magento Commerce versions) on AWS. The most appropriate choice depends on your requirements for cost, scale, availability, and flexibility as well as the AWS and Magento skills of your organization. This section outlines the full spectrum of deployment options along with some key characteristics to keep in mind when evaluating each option.

Table 1 – Summary of Magento Deployment Options on AWS

Deployment Style	Option	Description
Self-managed	Amazon Lightsail	Easy, cost effective start. Best for small business.
Self-managed	AWS Marketplace	AMI based solutions from variety of providers. Integrate other AWS services. Customer responsible for Magento patching/updates and configuring network

Deployment Style	Option	Description
Self-managed	AWS Quick Start for Magento	Deploys Magento reference architecture per AWS best practices in minutes. Highly configurable. Customer responsible for server and Magento maintenance.
Managed hosting	AWS Consulting partners	Easy deployments, including Magento store front customizations. Partner responsible for infrastructure and Magento maintenance. Best for organizations with less evolved IT departments or lacking skills/people to build/customize Magento deployments
Managed hosting	Magento Commerce Cloud	Most popular, highly scalable hosting option with Magento application and infrastructure managed by Magento and store front customizations managed by merchant.

Self-managed

AWS provides three options to quickly get started with a self-managed deployment of Magento. By self-managed we mean that AWS or an AWS partner provides the scripting or Amazon Machine Image (AMI) to deploy Magento and the necessary infrastructure dependencies, such as EC2 instances, in your AWS account. Once deployed, you are responsible for managing the deployment going forward including monitoring, patching, and upgrading.

Amazon Lightsail

Amazon Lightsail provides virtual servers, storage, databases, and networking that are easy-to-use and designed to allow you to quickly get started with the cloud. Popular application stacks, or blueprints, are available that can be deployed on Lightsail virtual servers. These stacks come preconfigured with all of the necessary components to get started with an application in minutes. The Magento application stack is provided by Bitnami and includes Apache, Varnish, Memcached, MySQL, and Magento Community Edition bundled in an AMI.

From the Lightsail console in your AWS account, you simply select the AWS Region and availability zone where you want to launch Magento, choose an instance plan, and launch your instance. After a few minutes, your instance is deployed and ready to access. Although Magento and all of its dependencies come preconfigured, you can still securely access the instance using a browser-based SSH interface or your favorite SSH

client to make lower level changes. The web-based Magento Administration user interface can also be used to create and customize your stores.

With the simplicity and cost effectiveness of deploying Magento through Lightsail also comes some important tradeoffs. These tradeoffs are important to keep in mind when it comes to scalability, availability, and maintenance of your e-commerce site. First, the Magento AMI provided by Bitnami that is used by Lightsail installs Magento and all dependencies on a single instance. Although this keeps the deployment simple, it also limits the ability to scale your e-commerce site, creates multiple single points of failure (SPOF), and leaves you with the responsibility to patch and update dependencies such as Memcached and MySQL. Therefore, selecting Lightsail as a deployment option should only be considered for smaller e-commerce sites where you expect a consistently low level of traffic from visitors.

AWS Marketplace

The AWS Marketplace is an e-commerce site where AWS customers can discover, procure, and deploy solutions provided by AWS partners. Thousands of solutions are available across more than 1,000 categories including infrastructure, business applications, machine learning, and many others. Deployment options supported on the Marketplace include applications deployed directly into customer accounts via AMIs or Docker containers, Amazon SageMaker, or SaaS solutions. All software purchased through the Marketplace appears on the customer's AWS invoice along with any other AWS resources consumed.

Magento deployment options currently available in the AWS Marketplace are AMI-based. These offerings include Magento and all the necessary dependencies such as MySQL, a web-server, and caching components. Therefore, AWS customers can deploy Magento directly into their AWS account and get up and running with Magento in minutes.

Similar to the Lightsail option, the customer is responsible for patching and upgrading Magento and its dependencies. In addition, the customer is responsible for configuring the networking environment, or Amazon Virtual Private Cloud (Amazon VPC), within which Magento is deployed. Lastly, customers should closely investigate and understand the scalability and high availability characteristics of each option, the Magento version bundled with each option, and any included customizations such as enhanced caching or multi-store setups. For example, some are all-in-one bundles that are intended to be deployed on a single Amazon EC2 instance. Although this provides a simpler configuration and lower cost, it introduces several single-points-of-failure and

lacks the ability to take advantage of the cloud's elasticity and high availability capabilities.

AWS Quick Start for Magento

AWS Quick Starts are push-button deployments of software solutions that are built and maintained by AWS and AWS partners. Quick Starts leverage AWS CloudFormation templates to script all aspects of a deployment following AWS best practices and include the ability to be launched in an existing or new VPC. There is no charge for the Quick Start itself, however, licensing costs for Magento Enterprise and costs incurred for the AWS resources launched by the Quick Start are the responsibility of the customer. The Quick Start also supports Magento Community Edition.

AWS resources launched with the AWS Quick Start for Magento include Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic File System (Amazon EFS), Amazon Relational Database Service (Amazon RDS) for MySQL or Amazon Aurora, Amazon ElastiCache for Redis, and Elastic Load Balancing. The web server installed with Magento by the Quick Start is NGINX. Multiple Magento servers are deployed in private subnets across two AWS Availability Zones in an Amazon Virtual Private Cloud (VPC) to enhance availability. Customers can use an AWS Auto Scaling Group to automatically scale the number of servers up and down based on load as well as replace instances that become unavailable. Outbound access to the internet for the Magento servers is provided by an AWS managed network address translation (NAT) gateway. A bastion host is also deployed in a public subnet to allow the customer to access the servers using SSH.

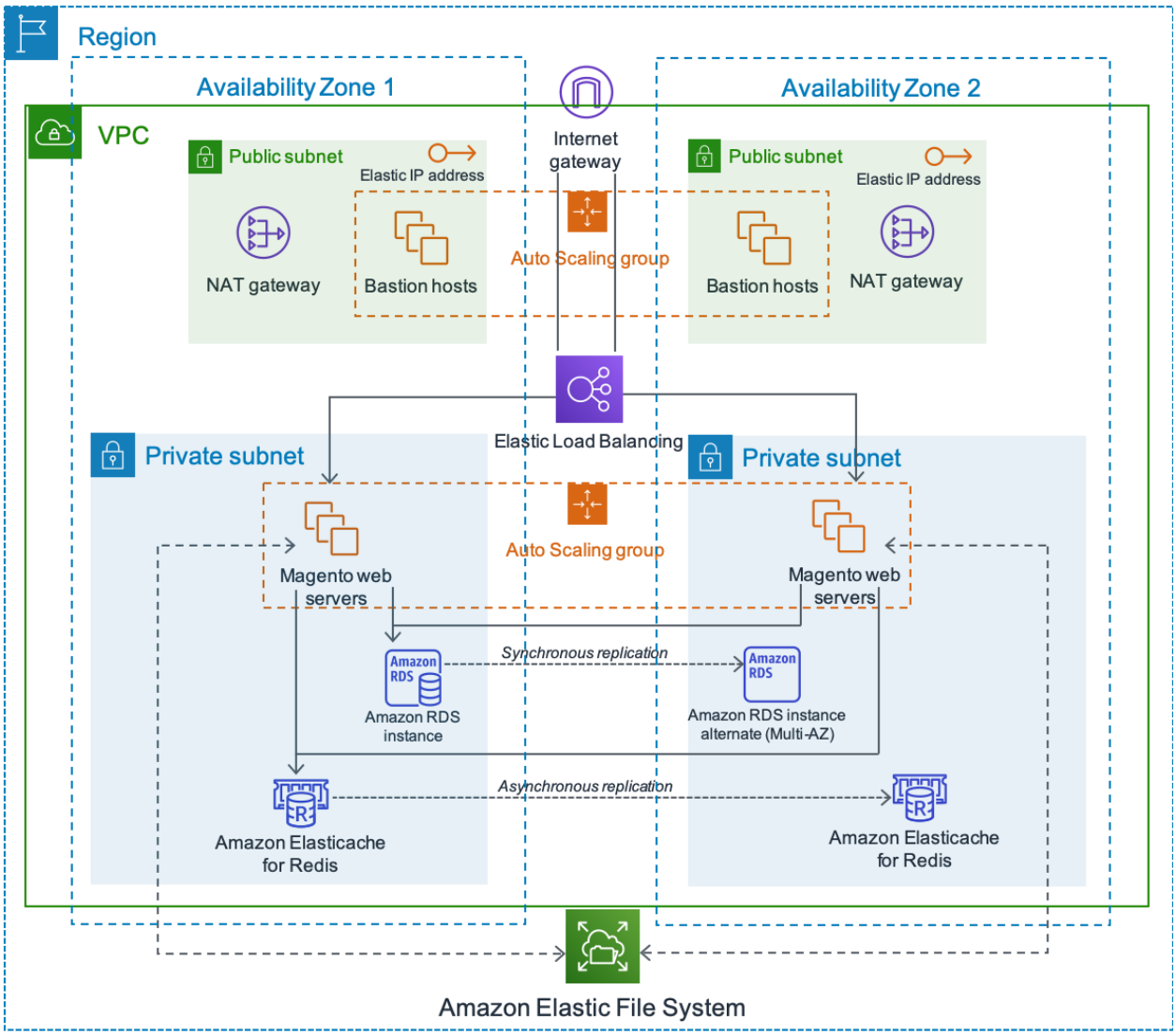

Figure 2 – Reference architecture deployed by AWS Quick Start for Magento

Managed Hosting

For customers who are not comfortable or do not have the resources to manage their own deployment of Magento on AWS, there are several companies that specialize in providing managed hosting deployments of Magento on AWS. These companies, such as CloudHesive, take care of the aspects of deploying, securing, patching, and maintaining Magento. Some also provide design services and custom development for Magento storefronts. You can use AWS Partner Finder to find and compare providers that specialize in Magento hosting.

Magento Commerce Cloud

One of the most popular managed hosting options for Magento on AWS is offered by Magento itself. Magento Commerce, part of Adobe Commerce Cloud, is a fully managed automated hosting platform for the Magento Commerce software. Magento Commerce Cloud comes with a variety of additional deployment and development features in addition to the self-hosted cloud Magento Commerce and Magento Open Source platforms.

Magento Commerce Cloud pre-provisioned infrastructure includes PHP, MySQL, Redis, RabbitMQ, and Elasticsearch technologies. In addition, it provides a Git-based workflow with automatic build and deploy for efficient rapid development and continuous deployment every time you push code changes in a platform as a service (PaaS) environment. Magento Commerce Cloud has AWS hosting that offers a scalable and secure environment for online sales and retailing.

Security and Compliance

Despite the common misconception that a cloud environment is less secure than on-premises infrastructure, strategic goals in relation to security and compliance are often key drivers for organizations to migrate to the cloud. Leading hyperscale cloud service providers, such as AWS, invest heavily in security and compliance and deliver a better security profile than what the biggest and most conservative organizations can deliver internally.

Security is a top priority at AWS. As an AWS customer, regardless of your size or investment, you inherit all the benefits of AWS experience, tested against the strictest of third-party assurance frameworks.

AWS Shared Responsibility Model

Under the AWS Shared Responsibility Model, AWS provides a global secure infrastructure and foundation for compute, storage, networking and database services, as well as higher level services. AWS provides a range of security services and features that AWS customers can use to secure their assets. AWS customers are responsible for protecting the confidentiality, integrity, and availability of their data in the cloud, and for meeting specific business requirements for information protection. In a simple way, AWS is responsible for security *of* the cloud and the customer is responsible for security *in* the cloud.

When leveraging the AWS Cloud, customers can choose a security solution that is designed to protect their organization's content, platform, applications, systems and networks, while also meeting their business needs. AWS offers a wide range of tools and features that help organizations increase privacy and control network access so they can more easily meet their needs within the AWS Shared Responsibility Model. Amazon Virtual Private Cloud (Amazon VPC) enables you to create a logically isolated portion of the AWS Cloud, from which you can launch Amazon EC2 instances in a virtual network that you define. Security groups allow you to define a virtual firewall around your EC2 instances, which contains rules that control the inbound and outbound traffic to your instances. Network access control lists (ACLs) provide an optional layer that allows you to control traffic in and out of one or more subnets in your VPC.

Best Practices

AWS and Magento offer a range of tools to help secure your cloud resources and to help you meet your compliance needs under organizational and industry standards. Best practices include the following:

Network Security

Amazon VPC allows you to create private networks within AWS and control network access to your instances and subnets. Use private or dedicated connectivity options such as AWS Direct Connect to connect your on-premises office/datacenter to AWS. If you are already using AWS in your organization then use AWS Private Link to connect Magento hosted cloud to your existing VPC. Lastly, always include a web application firewall (AWS WAF) and distributed denial of service (DDoS) mitigation technologies (AWS Shield) as part of your automatic scaling or content delivery strategy.

Data Encryption

Always encrypt your data both at rest and in transit. You can use TLS to encrypt the data in transit. Encryption at rest is achieved via data encryption capabilities available in AWS storage services such as Amazon Elastic Block Store (Amazon EBS), Amazon Simple Storage Service (Amazon S3), and Amazon Relational Database Service (Amazon RDS). Also, there are dedicated hardware-based cryptographic key storage options available for customers to help satisfy their compliance requirements.

Access Control

Protect your AWS and Magento user credentials. AWS credentials are used to access AWS services where Magento credentials are used to manage your storefront within

Magento. Use appropriate permissions across user accounts that access AWS resources and user accounts that access Magento store front customizing capabilities. AWS Identity and Access Management (IAM) enables you to create multiple users and manage the permissions. AWS supplies two types of security credentials: AWS access keys and X.509 certificates. Access keys and certificates for authentication to AWS services. As a good practice, it is recommended that you incorporate a key rotation mechanism into your application architecture. Lastly for extra security, AWS recommends that you use multifactor authentication (MFA) for all user accounts, including options for hardware-based authenticators, and integrate with federated identity providers such as on-premised corporate directories to reduce administrative overhead and improve end- user experience.

If users already have identities (user credentials) outside of AWS, such as in a corporate directory, then you can use identity federation along with AWS single sign on (AWS-SSO) to simplify the user management process. You can use the identity information from the external corporate directory system and use appropriate roles inside IAM for managing permissions.

AWS Secrets Manager can be used to rotate, manage, and retrieve database credentials, API keys, as well as Magento secrets (usernames/passwords). You can configure Secrets Manager to rotate secrets automatically, which can help you meet your security and compliance needs. Secrets Manager offers built-in integrations for Amazon Aurora Amazon RDS and can rotate credentials for these databases natively. To retrieve secrets for Magento application, you can replace plaintext secrets with a call to Secrets Manager APIs, eliminating the need to hard-code secrets in source code or update configuration files and redeploy code when secrets are rotated.

Monitoring and Logging

Always have the right monitoring and logging tools enabled to give you the visibility you need to spot issues before they impact your business. AWS features variety of services that give you deep visibility (who, what, when, and from where) such as 1) AWS CloudTrail for API calls (access requests), 2) AWS Config (configuration history), 3) Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health and 4) VPC flow logs that log all network traffic flowing through the VPC. In addition, you can configure CloudWatch Events to automatically alert as well as respond to adverse events. Lastly, Amazon GuardDuty is a threat detection service that automates continuous monitoring for malicious activity and unauthorized behavior using machine learning.

Security Guidance

AWS provides customers with guidance and expertise through online tools, resources, support, and professional services provided by AWS and its partners. AWS Trusted Advisor is an online tool that inspects your AWS environment to help close security gaps, and finds opportunities to save money, improve system performance, and increase reliability. AWS Advisories and Security Bulletins provide advisories around current vulnerabilities and threats, and enable customers to work with AWS security experts to address concerns like reporting abuse, vulnerabilities, and penetration testing. Magento security center provides advisories around latest Magento patches and security updates. Magento security scan tool, a free tool from Magento Commerce, can be used to monitor your websites for security risks, update malware patches, and detect unauthorized access.

PCI DSS

AWS supports a variety of security standards and compliance certifications including the Payment Card Industry Data Security Standard (PCI DSS), which is a crucial requirement for organizations who process credit cards and store cardholder information. Organizations that fail to comply with PCI requirements can expect large fines, which can also result in canceling their ability to process payments. PCI compliance requires organizations to safeguard their customers' payment card information following security requirements that include policies and procedures, software design, and network architecture.

AWS is certified as a PCI DSS 3.2 Level 1 Service Provider, the highest level of assessment available. Magento Commerce (Cloud) is PCI certified as a Level 1 Solution Provider. Organizations can use AWS and Magento's PCI Attestation of Compliance to aid their own PCI certification process.

The PCI DSS certification for an organization involves attestation of the following 12 requirements, broken into 6 groups: 1) Build and Maintain a Secure Network, 2) Protect Cardholder Data 3) Maintain a Vulnerability Management Program, 4) Implement Strong Access Control Measures, 5) Implement Strong Access Control Measures and 6) Regularly Monitor and Test Networks. For more information on PCI Compliance, refer to PCI DSS resources on AWS website and PCI Security Standards Council website.

AWS Architecture Components

As you plan your migration from an on-premises environment to AWS, you may find yourself introduced to new concepts. The first of those is the concept of managed services, which, simply defined, is a specific capability delivered as a service. For example, Amazon Elastic File System (Amazon EFS) provides network file system (NFS) based storage, eliminating the need to otherwise manage a fleet of EC2 instances to accomplish the same.

Another may be the scaling of shared services – so whereas before your on-premises environment may have consisted of a cluster of load balancing appliances, AWS offers (in a similar fashion as Amazon EFS described above) Elastic Load Balancing as a service, allowing you to manage the components used to deliver Magento in an end-to-end fashion.

Yet another concept may be the physical architecture of AWS – where a singular AWS Region comprises numerous groups of physically isolated datacenters (each group of data centers is referred to as an Availability Zone), providing native support for not just N+1 or active/passive or active/active resiliency, but doing so in physically distributed manner without implementing complex solutions.

In line with the above described services, where an on on-premises implementation of Magento may look like shared domain, networking, and load balancing services with an external content delivery network, all running on one or more servers (physical or virtual), in AWS, you may find yourself using any number of services to accomplish the same outcome from a service delivery perspective, while reducing cost and providing increased automation capabilities and increased scalability and resiliency capabilities.

Types of Services

Continuing the concepts described above, a typical collection of AWS services supporting your Magento environment, assuming no external or on-premises services are leveraged, might look like this:

- Amazon Route 53 for DNS
- Amazon CloudFront for content delivery network
- Amazon S3 for static content storage
- Elastic Loading Balancing through an Application Load Balancer for load balancing

- AWS Auto Scaling Groups of Amazon EC2 (virtual machine) instances for the Magento execution environment
- Amazon Elastic File System (Amazon EFS) for shared configuration/content storage
- Amazon Relational Database Service (Amazon RDS) for the Magento database environment
- Amazon ElastiCache for Redis for session and configuration caching
- Amazon Simple Email Service (Amazon SES) as your mail gateway

The above list, while detailed with regard to the Magento Environment itself, excludes numerous services providing supporting infrastructure of your platform (such as Amazon Virtual Private Cloud [Amazon VPC]) and as such this list should not be treated as a definitive list of services or requirements. Its intention is to illustrate the baseline AWS services you can leverage to host your Magento environment and eliminate unnecessary care and feeding of servers.

Monitoring

An additional benefit to leveraging managed services is that each service provides observability via metrics (CloudWatch Metrics), logs (CloudWatch Logs) and events (CloudWatch Events) providing the often-sought single pane of glass around each of your Magento environments. This data can include performance, exceptions (both technical measures) as well as business measures such as cost per transaction (via the CloudWatch API). This robust set of data, in addition to being available historically for reporting and near real time for dashboarding is also available for alerting – of both individuals in your organization and automations for self-healing processes.

From an organizational operations perspective, each of the preceding services also support the application of arbitrary metadata (referred to as *tags*), which can be used to allocate costs (cost per environment, as an example), in conjunction with AWS configuration management service, AWS Config, as well as an AWS robust audit trail solution, AWS CloudTrail, which provides insight into user or service access and change to AWS services.

Finally, services such as Application Load Balancer integrate into AWS X-Ray, which is the AWS distributed application tracing solution which provides for additional instrumentation into the execution environment of Magento and associated code.

DevOps

Access to on-demand compute and managed services, along with a common set of operational capabilities (authentication, authorization and auditing via AWS Identity and Access Management and CloudTrail, APIs and SDKs available on a variety of platforms and common metric, log and event ingestion services) support's an organization's goal towards automation.

AWS provides numerous services that abstract procedural level automation of resource lifecycle management including AWS CloudFormation, AWS Elastic Beanstalk, AWS OpsWorks, Amazon Elastic Container Service (Amazon ECS), and Amazon Elastic Kubernetes Service (Amazon EKS), in addition to a rich ecosystem of third-party services such as Terraform, Ansible and others.

There are multiple approaches with regards to managing a code base running in a compute environment. One such approach is using Amazon Machine Images (AMIs) where a machine image is programmatically configured at launch and cycled through with each code revision. Another approach is to statically maintain and configure managed machines leveraging agent or agent-based code deployment solutions. Lastly, container-based code deployments where a container is moved through different environments instead of code.

Each of these have their own strengths and weaknesses, however, container-based code deployments are becoming more common and support additional compute services such as AWS Fargate, a serverless container solution.

Regardless of the approach, the primary objective should be to maintain the compute environments in a hands-off approach with best security practices and monitoring solutions that provide good operational support.

For Magento, the management of the application lifecycle involves many of the same concepts, considered best practices and leveraged by other development platforms such as the use of a code repository and versioning (Git), separate build environments (for code compilation) and separate runtime environments (e.g. development, production. In addition, Magento supports the management of stores, cron entries etc. via the command line. This management logic can be built into the deployment pipeline. Also the configuration files can be versioned similar to code. Magento recommends management of assets via the same code repository as the application and configuration files, but this may create additional complexity, depending on the size of your application's assets. As an alternative approach, you can use one-way

synchronization between environment specific Amazon Elastic File System (Amazon EFS) volumes.

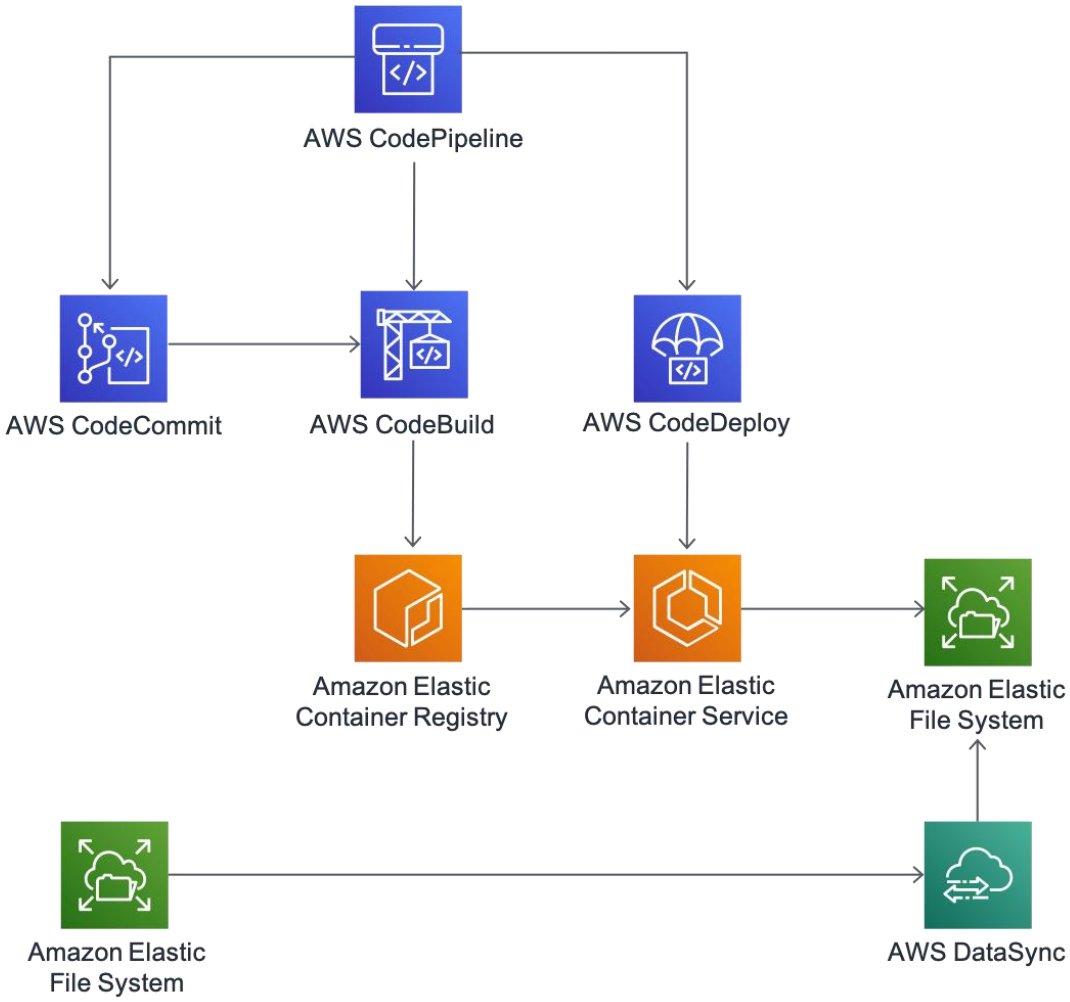

Figure 3 – Reference architecture for DevOps using containers

Connectivity

The migration of Magento may be part of an enterprise cloud adoption strategy, driven by a compelling event or a combination of the two. Regardless of the reasons, make sure to consider systems dependent on Magento and the systems Magento depends on. Connectivity between the Magento systems running in AWS and these dependent systems (whether in AWS, a data center, or a SaaS/PaaS provider) is key to planning and successfully executing a migration. Make sure to also consider connectivity for Magento admin, System admin, Database admin and Infrastructure admin.

Types of Connectivity

Five connectivity channels are available in AWS:

- API (serverless resources such as Amazon S3, Amazon DynamoDB, etc.)
- Internet (publicly routed, typically brokered via SFTP based automation)
- VPN (IPSEC/B2B)
- AWS Direct Connect (Dedicated connectivity, via either a physical or logical connection)
- Intra VPC (typically environment:tier <-> environment:tier),
- Inter VPC (via Peering or Transit Gateway) and AWS PrivateLink (typically private connectivity, AWS based SaaS offerings).

Whether your connectivity is between tiers, environments, other services within your enterprise on and off AWS or with a partner, each of these approaches has advantages and disadvantages associated with them.

Network Dependencies

Of most importance, however, is fully understanding the network dependency map that exists within your Magento environment ahead of migration. Having a dependency map can help better plan the steps required in the migration and avoid accelerated post migration changes required to account for missed dependencies, both building to support a successful migration.

Investigation around this might begin with reviewing load balancer and firewall configurations as they relate to host names and IP Addresses of the servers Magento is running on, the Magento and related service configurations and any associated logs – particularly load balancer, firewall, and application performance management (APM) logs which may provide hints around third-party services not otherwise discoverable.

Service Congruency

For example, a typical Magento environment is likely accessed from the Internet via a Load Balancing appliance or a Firewall providing NAT Public IP Address space, and may have integrations to other internet-based services (e.g. a web service offered by a payment processor or shipping company) as well as legacy integrations via SFTP or a similar internet-based file sharing service.

These legacy integrations described earlier would translate into AWS services in the way of an Application Load Balancer on one set of subnets protected via a Security Group and Web Application Firewall, to a fleet of EC2 instances running Magento dependent services. A NAT gateway providing egress access to internet based services. Same or separate fleet of EC2 instances for providing integration with other internet services such as payment processors, or leverage serverless solutions such AWS lambda and AWS Transfer for SFTP for Amazon S3 based on the use-case.

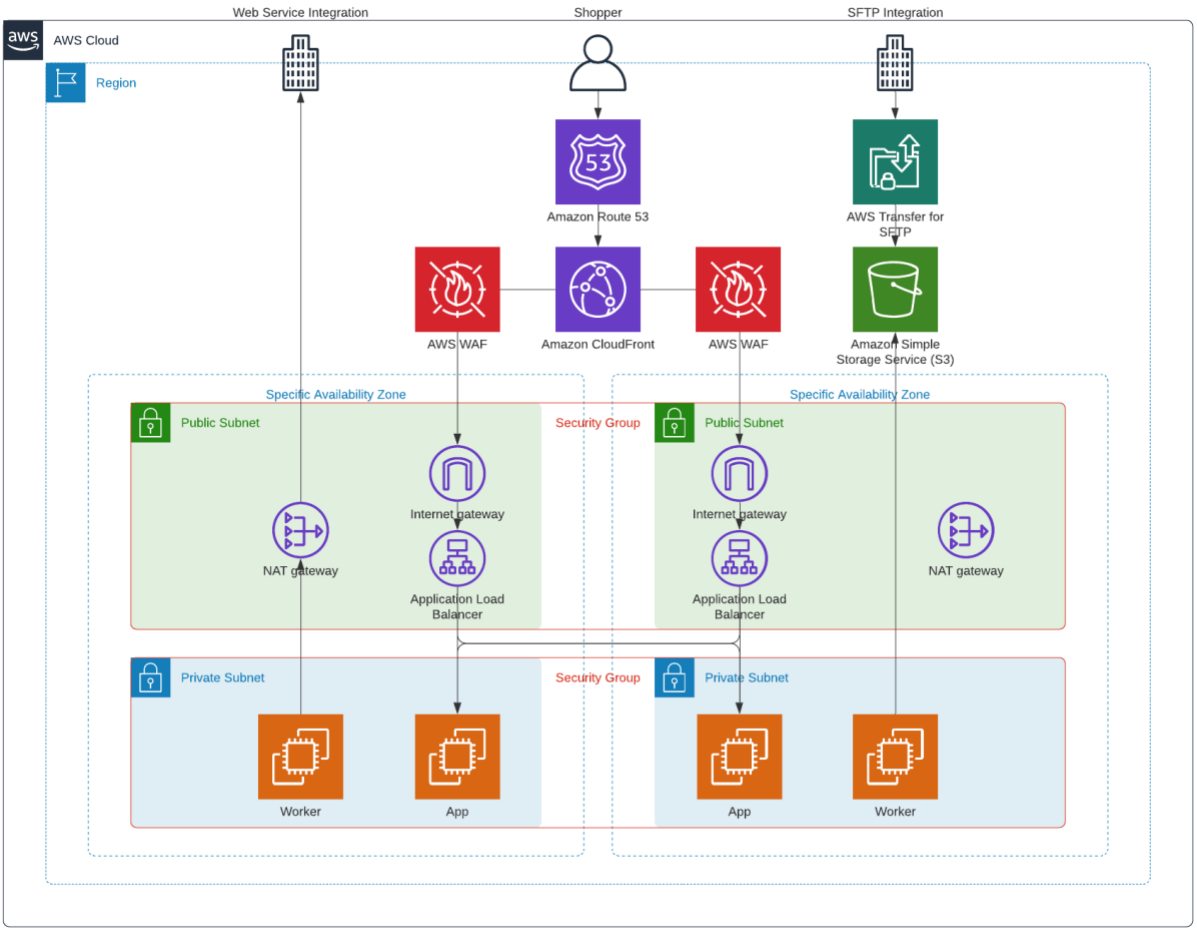

Figure 4 – Reference architecture showing service congruencies for containers

Migration Plan

This section steps through an example migration, broken into four phases:

Planning

Planning in migration involves identifying goals, scope and business requirements. Often overlooked, and related to these items, are having quantifiable goals as measures of success, specifically with regard to performance (e.g. performance shall remain the same or be improved by x margin) as well as a full understanding of the components that comprise the to-be-migrated environment(s).

To solve for the first, you can use the existing monitoring used in your enterprise or use AWS native monitoring capabilities. Rather than focusing on machine level statistics (CPU, memory, storage consumption, storage IO, network consumption, etc.), which are important with regard to machine configuration, focus on end-user experience – e.g. response times for key activities, such as adding to cart, checking out, payment, etc.

To solve for the second, you can use existing financial records, inventories, or monitoring to identify the components that comprise each environment, including those that may not be evident (e.g. virtualized or delivered as a third-party solution, such as DNS or CDN).

Staging

After you have identified measures of success and completed the discovery, you should have a sense of the baseline services required to support your application. Compared to your existing AWS footprint (if there is one) you may opt to create an entirely new AWS Account or Amazon VPC, use an existing AWS Account or Amazon VPC, or go through the collective supporting characteristics for first time AWS adoption found in the AWS Landing Zone Solution.

Once this work is complete, you can begin prototyping your environment on AWS – familiarizing yourself with some of the services mentioned above and identifying potential replacements to share service, appliance or server based services. This is the re-platforming approach.

Once this is setup, you can begin to stage an environment for your first test migration. This environment would essentially replicate your current production environment from a software, configuration, code, and management perspective and aim to test both

basic functionality of the platform as well as your data backup/restore and/or data replication processes that will be leveraged during the actual migration.

This would be your second decision point for rehosting versus re-platforming: there are multiple approaches you can take to migrating your servers to AWS – from as basic as exporting a virtual machine and importing it into AWS, to using CloudEndure Migration for real time replication of the machine. While opinions vary on the subject, the reader is strongly encouraged to take an approach biased to re-platforming as it presents an opportunity to significantly improve operations of the platform with minimal (but not nonexistent) up front work.

This first attempt at the migration provides you with a harness to execute, record, measure and modify the previously defined plans and tests, understand baseline performance as well as capture step-timings and improve the overall cutover plan. This step can be repeated as often or as frequently as needed until a comfort level is obtained to move into the next step.

Cutover

The final step of the migration process is the live migration from the current on-premises environment to the AWS based environment. Essentially the process is similar as the previous step, though you will likely place the store in maintenance mode and/or use a static maintenance site to prevent live transactions during the most critical parts of the cutover.

An important part of this step is to have agreement with all participating parties around the maximum time to be spent on each step as well as critical milestone steps having pass/fail criteria along with rollback criteria in the event of a failure. Having these steps defined ahead of time avoid last minute decisions based on arbitrary criteria from being made in the event an unanticipated challenge is met (gamedays, tabletop exercises or simulations attempt to drive a similar thought process).

At this point you've effectively migrated your Magento platform to AWS, and you can begin the decommissioning process – first through soft methods (e.g. stopping services) on to harder methods (decommissioning of virtual machines and servers).

The cutover process varies based on the specific Magento installations architecture, but in general, you want to stop existing Magento processes as well as disable any scheduled cron jobs.

You can redirect traffic by changing the current content delivery network (CDN) configuration, the current DNS configuration (ensuring as part of premigration you've

lowered the TTL) or, optionally, using the EC2 instances on AWS as new upstream targets for your current load balancer (this being the least preferred option).

Some customers may use this as an opportunity to change DNS or CDN providers and these migration activities can be performed as part of a pre-migration or post-migration, following their own migration process and steps.

Optimization

Previously made architecture decisions can also be revisited post-migration. For example, if you used a rehost approach, you can evaluate and re-platform post-migration services, or right-size services based on current and forecasted load. This period of time is an excellent period to evaluate potential opportunities for reserved instance purchases, which offer a discount to service cost in trade for commitment to use a service for a period of time (ideal for Amazon RDS, Amazon ElastiCache and permanent Amazon EC2 instances in production).

Other Areas of Consideration

Magento M1 to M2 Migration

Magento 1.x versions (M1) to Magento 2.x versions (M2) migration consists of four main components:

1. Data Migration,
2. Extensions Migration,
3. Custom Code Migration
4. Theme migrations

A deep dive into these four areas specific to how the organization has setup M1 outlines the challenges and the migration strategy. Magento has developed the Data Migration Tool for data migration and Code Migration Toolkit for code migration. These resources are described in detail in the Magento Migration Guide.

Magento Hardware Sizing Guidelines on AWS

As per Magento hardware recommendations, Magento recommends that one CPU core can effectively serve around two (up to four) Magento requests along with one cron process simultaneously. Determine your organization’s stable expected request rate to

find the appropriate number of cores needed to support your application and use automatic scaling to dynamically extend web tier nodes as needed.

N[Cores] = (N[Expected Requests] / 2) + N [Expected Cron Processes]

In addition, Magento recommends at least 2 GB memory on build servers and 1 GB on web nodes along with sufficient network bandwidth to prevent bottlenecks on read-write operations.

Keeping these principles in mind, choose the appropriate instance from the Amazon EC2 instance types that balances your organization's cost and business needs. Instance types from the Amazon EC2 general purpose family (especially M types) provide a good balance between compute, memory, and network resources for Magento applications.

Backups and Disaster Recovery

As a best practice, explore backup and disaster recovery options based on your business needs. Backup and disaster recovery options depend on the deployment option and database choices (Amazon RDS vs Amazon Aurora) you choose. Your organization's business needs dictate the recovery point objective (RPO) (i.e. maximum time to last backup and the recovery time objective (RTO). RTO often varies depending upon the size of your organization's storage.

At a minimum, we recommend that you take regular database backups and have a working copy of the AWS CloudFormation template to provision the infrastructure when needed in case a recovery situation arises. Alternately, you can choose to have a working Amazon Machine Image (AMI) copy that has Magento installed along with your organization's latest changes or customizations. You can use this AMI to provision infrastructure using an AWS CloudFormation template to reduce the overall RTO.

Conclusion

This paper presented the business drivers for migrating Magento to the AWS Cloud along with the strategies and considerations. Migrating Magento eCommerce software on AWS provides a secure and scalable foundation for delivering great digital experiences for customers. As you prepare for your Magento migration to AWS, we recommend that you consider the guidance outlined in this document and consult the additional references provided in the following Further Reading section.

Contributors

Contributors to this document include:

- Anuj Ratra, Sr. Solutions Architect, Amazon Web Services
- James Jory, Sr. Solutions Architect, Amazon Web Services
- Patrick Hannah, Chief Technology Officer, CloudHesive

Further Reading

For additional information, see:

- AWS Partner Finder
- Magento Developer Documentation
- Magento on AWS Quick Start
- AWS Digital Customer Experience Competency & Partners
- AWS Webinar: How to Manage Organizational Change and Cultural Impact During a Cloud Transformation
- Magento Hardware Sizing
- Magento Configuration Best Practices
- Migrating to AWS: Best Practices and Strategies

Document Revisions

Date	Description
April 2020	First Publication

Notes

1 https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-322.pdf

2 https://www.gartner.com/en/documents/1485116/migrating-applications-to-the-cloud-rehost-refactor-revi