Operational Checklists for AWS

Steve Morad
February 2012

(Please consult http://aws.amazon.com/whitepapers/ for the latest version of this paper)

Table of Contents

Abstract

Deploying an application on Amazon Web Services (AWS) is fast, easy, and cost-effective. Before deploying a cloud application in production, it is useful to have a checklist to assist in evaluating your application against a list of essential and recommended best practices. This paper highlights some of these operational and architectural considerations that you should consider as you deploy applications on AWS.

Introduction

Amazon Web Services (AWS) is a flexible, cost-effective, and easy-to-use cloud computing platform. AWS provides a suite of infrastructure services that you can use to deploy your applications. To get the maximum benefit out of the cloud platform, we recommend that you leverage AWS's complimentary services and follow the best practices. Organizations that invest time and resources assessing the operational readiness of their applications before launch have a much higher rate of satisfaction than those who don't. When performing this work, checklists can be an invaluable mechanism to ensure that applications are evaluated consistently and holistically.

The level of operational assessment will vary depending on the organization's cloud maturity and the application's development phase, availability needs, and data sensitivity requirements. This paper provides two checklists to support these varying assessment needs:

- **Basic Operations Checklist** - covers common high-level technical questions that organizations should consider as they adopt different AWS services and are planning for a launch.
- **Enterprise Operations Checklist** - provides a more in-depth operational review of suggested best practices that an enterprise should consider when developing a mature cloud strategy.

This paper is targeted at developers and architects who are looking for operational and architectural guidance from AWS to help assess their application's operational readiness. These individuals typically support enterprise organizations developing formal cloud strategies or performing formal technology reviews. However, it could also be useful to any organization for comparing its planned use of AWS against these essential and recommended best practices.

How to Use the Checklists

Basic Operations Checklist - This checklist can be used to evaluate your application before you launch it in production on AWS. It includes the typical questions that AWS Solution Architects ask customers when they seek guidance to avoid common pitfalls not obvious to new users. When each item is checked off with a satisfactory and affirmative answer, you can confidently deploy your applications in the cloud. Checklist items are designed such that it will instigate the right conversations about whether or not the specific service or concept is applicable to your application and, if so, whether or not it has been adequately addressed. We plan to update this checklist when new application services are launched or when new best practices are identified.

Enterprise Operations Checklist - This checklist is intended to help enterprises think through various operational considerations as they deploy sophisticated enterprise applications on AWS. It can also be used to help you build a cloud migration and operation strategy for your organization. This section is also further divided into two parts. The first part provides a high-level checklist with brief descriptions for each operational consideration. The second part provides more detail about each checklist item, as well as links to additional information.

Checklist	Intended Usage	Target Customer
Basic Operations Checklist	To help customers assess their application's use of specific services and features before they launch	Developers and system architects
Enterprise Operations Checklist	To assist enterprises in identifying key items to think about as they build a cloud migration and operational strategy	Enterprise architects

Basic Operations Checklist

	Checklist Item
☐	We use AWS Identity and Access Management (IAM) to provide user-specific, rather than shared credentials for making AWS infrastructure requests. Learn more…
☐	We understand which of our instances is Amazon Elastic Block Store (Amazon EBS)-backed versus instance store-backed, have intentionally chosen the most appropriate type of storage, and understand the implications to data persistence, backup and recovery. Learn more…
☐	We understand AWS dynamic IP addressing and have ensured that our application will function when application components are restarted (e.g., using 3rd-party or Elastic Load Balancing, Amazon Virtual Private Cloud (Amazon VPC) static address assignments, elastic IP addresses, or dynamic DNS). Learn more…
☐	We use separate Amazon EBS volumes for the operating system and application/database data where appropriate. Learn more …
☐	We regularly back up our Amazon Elastic Compute Cloud (Amazon EC2) instances using Amazon EBS snapshots or another 3rd-party backup tool. Learn more …
☐	We have a fully tested plan for recovering our Amazon EC2 instances or Amazon EBS volumes when they fail, either through customized "golden" Amazon Machine Images (AMIs), Amazon EBS snapshots, boot-strapping, or using our own backup and recovery tools. Learn more…
☐	We have deployed critical components of our applications across multiple availability zones, are appropriately replicating data between zones, and have tested how failure within these components affects application availability. Learn more…
☐	We understand how fail-over will occur across application components deployed in multiple availability zones and are using 3rd-party or Elastic Load Balancing and elastic IP addresses where appropriate. Learn more…
☐	We have a plan for patching, updating, and securing our Amazon EC2 operating system, applications, and customized AMIs. Learn more…
☐	We use appropriate operating system user account access credentials and are not sharing the AWS instance key pair private key with all systems administrators. Learn more…
☐	We have implemented secure Security Group rules and nested Security Groups to create a hierarchical network topology where appropriate. Learn more…
☐	We use "CNAME" records to map our DNS name to our Elastic Load Balancing or Amazon Simple Storage Service (Amazon S3) buckets and NOT "A" records.
☐	Before sharing our customized Amazon Machine Images with others, we removed all confidential or sensitive information including embedded public/private instance key pairs and reviewed all SSH *authorized_keys* files. Learn more…
☐	We have fully tested our AWS-hosted application, including performance testing, prior to going live. Learn more…

Enterprise Operations Checklist

For each checklist category in the table below, additional details are provided through internal references to subsequent sections of this document.

	Checklist Category	Description
☐	Billing & Account Management	Has your organization developed an approach for billing and account management? Has your organization determined whether or not multiple accounts will be used and how billing will be handled?
☐	Security & Access Management	Has your organization developed a strategy for managing AWS API, console, operating system, network, and data access?
☐	Asset Management	Does your organization have a strategy for identifying and tracking AWS provisioned resources?
☐	Application HA/Resilience	Does the implemented AWS solution meet or exceed the application's high availability and resilience requirements?
☐	Application DR/Backup	Does the implemented AWS solution meet or exceed the application's disaster recovery (DR) and backup requirements?
☐	Monitoring & Incident Management	Has your organization instrumented appropriate monitoring tools and integrated your AWS resources into its incident management processes?
☐	Configuration & Change Management	Does your organization have a configuration and change management strategy for its AWS resources?
☐	Release & Deployment Management	Has your organization determined how it will integrate application releases and deployments with its configuration and change management strategy?

Additional Checklist Information

The following subsections provide additional details and considerations for each checklist category in the table above.

Billing & Account Management

Does your organization have a strategy for managing AWS billing and accounts? An effective strategy would include how an organization will handle multiple AWS accounts, billing, and charge back. At a minimum, an organization's billing and account management strategy should be able to answer the following questions:

- Will more than one AWS master account be necessary?
  Customers utilize multiple AWS accounts for different reasons, including security segregation and increased billing or charge back granularity. Consolidated billing accounts can be used to aggregate billing from multiple accounts; however, this approach increases the administrative overhead associated with managing and sharing resources across multiple accounts.
- What is the purpose of each account and how will they be linked?
  Organizations can simplify the use of multiple accounts, by leveraging a single, consolidated billing account for billing purposes and sub or linked accounts for consuming AWS resources. Sub accounts can then be used for different purposes such as the separation of dev/test/prod or for the creation of completely separate environments for various business units or customers.
- Has your organization requested consolidated or invoice billing?
  Consolidated billing allows customers to receive a single bill for multiple AWS accounts and to potentially lower costs by rolling up usage across these accounts. Invoice billing allows AWS customers to receive their AWS bills through invoices rather than on a corporate or personal credit card.
- What form of chargeback is required and how will chargeback rates or bills be calculated?
- What billing optimization steps will be taken?
  Customers can optimize their costs on AWS by choosing appropriate instance sizes, automating their environments to scale up or down depending on utilization or schedule, and leveraging the most appropriate pricing model (on demand, reserved, or spot instances).
- How will the organization leverage reserved or spot instances?
  Please see the Amazon EC2 Instance Purchasing Options website for descriptions and recommendations associated with each purchasing option.

Additional information related to AWS Billing:

- Consolidated Billing Guide
- Amazon EC2 Instance Purchasing Options
- AWS Sales and Business Development
- Add Amazon CloudWatch Billing Alarms
- How AWS Pricing Works

Security & Access Management

Security and access management is extremely important to AWS and our customers. An organization should review and incorporate the following resources into their access management strategy:

- AWS Security and Compliance Center
- Overview of Security Processes Whitepaper
- Risk and Compliance Whitepaper
- Security Best Practices

AWS API Credential Strategy

Does your organization have a strategy for managing and leveraging the rich set of access credential options provided by AWS? An effective strategy would include the consideration of when your organization will use, and how it will manage, AWS Identity and Access Management (IAM) users, symmetric access keys, asymmetric X.509 certificates, console passwords, and hardware or virtual multifactor authentication devices.

At a minimum, an organization's AWS credential strategy should answer the following questions:

- How will your administrators, systems, or applications authenticate their AWS infrastructure requests to AWS APIs?

  AWS provides a number of authentication mechanisms including a console, account IDs and secret keys, X.509 certificates, and multi-factor authentication (MFA) devices to control access to AWS APIs. Console authentication is the most appropriate for administrative or manual activities, account IDs and secret keys for accessing REST-based interfaces or tools, and X.509 certificates for SOAP-based interfaces and tools. Your organization should consider the circumstances under which it will leverage access keys, x.509 certificates, console passwords, or multifactor authentication devices.
- Has your organization established internal credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials?

  Incorporating AWS access credentials into an organization's existing internal credential management policies and procedures is an important and typically straightforward exercise for our customers.
- Is your organization leveraging IAM users and/or tokens?

  AWS recommends leveraging AWS Identity and Access Management credentials with internal security processes and controls for controlling unique, role-based, least privilege access to AWS APIs.
- Will any AWS credentials be embedded in AMIs?

  AWS credentials can be provided to AWS aware applications through the Amazon Machine Image used to launch the application. When this approach is taken, customers should intentionally incorporate the management of these credentials in their image and instance configuration management processes.
- Has your organization segregated IAM administrative privileges from regular user privileges?

  AWS recommends that organizations segregate security credential administration from standard administrative privileges by creating an IAM administrative role and restricting IAM actions from other compute, storage, and networking roles.

Additional information related to AWS credentials:

- Introduction to AWS Security Credentials
- Amazon Identity and Access Management
- Making Secure Requests to Amazon Web Services

Amazon EC2 Instance Credential Strategy

Does your organization have a strategy for managing OS and application credentials for authentication, authorization, and audit tracking? An effective strategy would include the consideration of how user access will be instrumented, controlled, and monitored for your Amazon EC2 instances and applications. Under the AWS shared security model, your organization is responsible for OS and application level identity and access management. A full description of technologies and techniques to accomplish these responsibilities is beyond the scope of this document; however, customers have leveraged Lightweight Directory Access Protocol (LDAP), Active Directory, Web Application Firewalls, and other security technologies to control access to their Amazon EC2 instances, application, and data.

Network Access

An organization must determine whether it will use the Amazon EC2 "Public" environment, Amazon Virtual Private Cloud (Amazon VPC), or a combination of both. Connectivity to and from their AWS and corporate environments must be well understood and may leverage the Internet, hardware or software virtual private networks, or direct connections. Additionally, an organization should have a strategy for managing Amazon EC2 Security Groups, Amazon VPC network routing or access control lists, and host-based firewalls/intrusion detection systems if applicable. A comprehensive network access strategy will likely incorporate the following components:

- Use of Amazon EC2 "Public" and Amazon EC2 Virtual Private Cloud environments
- Use of network connectivity and controls between AWS and corporate networks
- Use of transport encryption protocols
- Use of operating system access controls including Amazon EC2 Security Group configuration, OS hardening, security patches, host-based firewall, intrusion detection/prevention, and monitoring software configuration.

Additional information related to network access:

- Amazon EC2 Network and Security
- Amazon Virtual Private Cloud
- Amazon VPC Security Groups
- Amazon Elastic Network Interfaces
- Elastic Load Balancing Security Features
- AWS Direct Connect
- AWS Security Best Practices

Data Access

An organization must determine how it will protect access to its data. AWS provides physical barriers to protect physical access to underlying storage media as well as a number of logical access controls that can be used by your organization to protect its data. Additionally, organizations should consider what level, if any, additional controls are required. Potential data access controls include the following:

- AWS logical access control lists or access policies
- Disk, file, or database-level encryption for data-at-rest
- Application or data-level access control, intrusion or leakage detection software
- Data lifecycle or records management tools

Additional information related to data access:

- Amazon S3 Access Control
- Amazon S3 Data Encryption
- Amazon EBS Snapshot Permissions
- AWS Security Best Practices

AWS Management Console Strategy

Has your organization thought through the various console options for system administrators using AWS? Examples include leveraging the AWS provided, out-of-the-box console, 3rd-party AWS consoles, or building a custom, internal console.

Organizations should ensure that their use of AWS fits within their existing IT management policies, procedures, and strategies. The AWS provided console has been successfully integrated into many of our customer's existing IT management practices; however, some organizations have leveraged 3rd-party consoles or built their own for tighter/automated integration into their change management, billing, or security systems.

Additional information related to AWS console:

- AWS Management Console
- Please contact AWS Sales and Business Development for a discussion of various 3rd-party consoles provided by our large network of solutions providers.

Asset Management

Does your organization have a strategy for tracking and managing its AWS deployed assets? An effective strategy would include whether or not an internal asset management system will be integrated with AWS and how AWS provided asset management capabilities will be leveraged. At a minimum, an organization's asset management strategy should be able to answer the following questions:

- Is your organization leveraging AWS provided instance and service specific metadata as part of its asset management strategy?
  AWS provides out-of-the-box metadata for each of its services to help your organization identify, track, and manage your AWS resources. Customers can leverage this metadata to track Amazon EC2 instances or storage by server image (AMI), operating system, compute architecture (32-bit or 64-bit), volume id, snapshot, attached storage, and many other categories.
- Is your organization leveraging custom resource tags to track and identify AWS resources?
  In addition to the out-of-the-box metadata, AWS allows customers to apply their own custom tags. Resources could be tagged by support team, application version, cost center, environment type, lifecycle status or any other category that will help your organization more effectively manage its AWS resource assets.
- Does your organization have a resource tagging strategy?
  Although AWS supports ad hoc resource tagging, an organization will get the most benefit from tagging if they strategically plan for the intentional and systematic use of resource tags.
- How will AWS assets be integrated with internal asset management systems?
  AWS resources can be programmatically or manually queried to easily pull service and resource metadata into existing asset management systems and processes.

Additional information related to asset management:

- Amazon EC2 Instance Metadata
- Using Amazon EC2 Tags
- Using AWS CloudFormation or Tagging Auto Scaling Groups to automatically tag resources
- Amazon S3 Object Key and Metadata

Application HA/Resilience

Every application has specific High Availability (HA) requirements and characteristics. AWS provides a number of infrastructure building blocks to help your organization meet these requirements cost effectively. At a high level, an effective HA strategy would include instance redundancy, use of multiple availability zones within a region, load balancing, auto scaling, monitoring, and recovery within a region. Critical applications should ensure that all single points of failure are identified and evaluated based on the application's availability requirements and risk profile. An effective HA strategy will include not only how a single component will recover (e.g. will a failed instance be automatically or manually relaunched in the same or different availability zone?), but also what HA testing will be performed to ensure that the application can be recovered as expected. The following are Amazon Web Services that your organization can consider leveraging for high availability:

- Running multiple Amazon EC2 instances in multiple Availability Zones
- Elastic Load Balancing for load balancing across multiple Availability Zones
- Auto Scaling for automated instance recovery or scaling
- Amazon CloudWatch Metrics (custom metrics as well as out-of-the-box) and Alarms
- Multi-AZ Amazon RDS for multiple Availability Zone managed databases
- Elastic IP Addresses for static IP addresses that can be remapped between instances
- Amazon EBS Snapshots for point-in-time snapshots of Amazon EBS volumes

- Storing objects in Amazon S3 and/or using Amazon CloudFront for distribution
- Storing key/value pairs in Amazon SimpleDB or Amazon DynamoDB
- AWS Elastic Beanstalk for environment and application version management
- Leverage synchronous data replication technologies like database mirroring
- Reserving HA capacity through Amazon EC2 Reserved Instances

Additional information related to designing highly available applications in the cloud includes the following:

- Designing Fault-Tolerant Applications in the Cloud
- Architecting for the Cloud: Best Practices
- RDBMS in the Cloud: Microsoft SQL Server 2008 R2
- MongoDB on AWS
- Storage Options in the Cloud
- Cloud Architectures

Application DR/Backup

Every application has specific Disaster Recovery (DR) requirements that should be tied to recovery point and recovery time objectives as well as any geographic requirements that restrict the physical proximity between primary and disaster recovery sites. An effective DR strategy will include not only how a single component will recover (e.g. will a failed region be automatically or manually re-launched in another region using cold, warm, or hot stand-bys?), but also what DR testing will be performed to ensure that the application can be recovered as expected. At a high level, an effective DR strategy would include regional redundancy, global traffic management or load balancing, monitoring, and region-to-region recovery. The following are Amazon Web Services and techniques that your organization can consider as part of a DR strategy:

- Store data, Amazon Machine Images, or run additional instances in multiple AWS regions
- Use Amazon Route 53 for DNS-based regional fail-over
- Leverage asynchronous data replication technologies like database log shipping
- Reserve DR capacity in another region through Amazon EC2 Reserved Instances

Additional information related to designing cloud applications for DR includes the following:

- Designing Fault-Tolerant Applications in the Cloud
- RDBMS in the Cloud: Microsoft SQL Server 2008 R2
- Using AWS for Disaster Recovery

Monitoring & Incident Management

Application and operating system monitoring is essential for any organization to ensure that it can effectively detect and respond to incidents. At a high level, an effective monitoring strategy will ensure that monitoring tools are instrumented at the appropriate level for an application based on its business criticality. An effective incident management strategy will incorporate both AWS and customer monitoring data with its event and incident management tools and processes. An organization should consider the following tools as part of this strategy:

- Amazon CloudWatch out-of-the-box metrics for AWS monitoring, alerting, and automated provisioning
- Amazon CloudWatch custom metrics for Application monitoring, alerting, and automated provisioning
- Amazon EC2 instance health for viewing status checks and scheduled events for your instances
- Amazon SNS for setting up, operating, and sending notifications from the cloud
- Operating system monitoring tools for OS-level monitoring
- Application monitoring tools for Application-level monitoring
- Simulated transaction monitoring tools for end-to-end system monitoring

Additional information related to monitoring and incident management includes the following:

- Amazon EC2 Troubleshooting
- Monitoring Amazon EC2 instances

Configuration & Change Management

Has your organization determined how it will integrate cloud configuration and change management into its IT operational processes? Intentional, controlled change management is essential whether an organization has mature IT Service Management processes, fully implemented Information Technology Infrastructure Library (ITIL), or has yet to create a change management database (CMDB). An effective cloud configuration and change management practice would include cloud concepts like the following:

- How will your organization manage server images (AMIs)?
  Server images must be periodically updated with patches and software updates. AWS provides a number of tools that can be incorporated in your organization's image management processes to assist in the creation and management of AWS images.
- Will instances be automatically configured at launch or manually configured later?
  Automating instance configuration on boot, by passing user-data to the instance on boot or embedding change and configuration management agents in a server image, allows instances and applications to take advantage of instance meta-data, cloud automation, scaling, and high-availability capabilities.
- How will OS credentials be instrumented and controlled when instances are launched or terminated?
  Typically, organizations preconfigure their server images to automatically connect and register with corporate LDAP or Active Directory domains when they are launched to provide centralized OS credentials management and control.
- How will patches and upgrades be applied?
  Organizations take different patch and upgrade management approaches depending on their application's characteristics and requirements. Updates can be applied to existing instances using traditional software deployment tools or by replacing outdated software running on older instances with newer, patched, and upgraded server images.
- Will applications be managed as homogeneous fleets?
  Managing applications as homogeneous fleets allows infrastructure to be dynamically and automatically provisioned or released based on predictable utilization patterns.
- How will your organization manage changes to OS hardening baselines, configure security groups or OS firewalls, and monitor their instances for intrusions or unauthorized changes?
  Most organizations already have existing internal IT change and configuration management processes and tools that can incorporate AWS related changes with minimal modification.

Release & Deployment Management

How will your organization integrate application releases and deployments with its change and configuration management strategy? In the cloud, the traditional lines between infrastructure changes and application deployments can be blurred, if not completely erased. In addition to traditional code development, testing, and versioning concepts, organizations should also consider the following cloud integration points for application releases or deployments:

- What software release and deployment process or methodology will your organization leverage?
  Organizations have full control over their software release and deployment processes. Some organizations utilize traditional release and deployment processes that deploy approved releases from a controlled software repository to existing servers. Other organizations bundle, promote, and release complete software stacks incorporating applications and server images combined throughout the development lifecycle.
- Will newer versions of an application be phased-in to existing server farms and older versions phased-out?
  AWS provides organizations with the opportunity to implement new, shorter maintenance window deployment models by quickly and cheaply spinning up new application versions to gradually replace older instances over time.
- Will weighted load distribution patterns be used to intentionally deploy, test, migrate, and roll-out or roll-back new application releases?
  Just as AWS enables organizations to take advantage of new deployment models, these same models can also be used as part of testing, migration, or roll-back processes to more quickly and seamlessly support release and deployment processes.
- How can your organization leverage infrastructure boot-strapping and application deployment tools to more quickly and effectively introduce or roll-back changes?
  Releasing and deploying applications on AWS provides organizations with an opportunity to reevaluate its existing processes to determine where they can improve efficiencies through cloud-friendly change, configuration, release, and deployment automation.
- How can your organization make its applications more infrastructure-aware so applications can become active participants in making the infrastructure changes required to support a specific software release or deployment?
  Traditional applications are dependent on coordination with completely independent infrastructure management teams and change control processes. Often, a deployment weekend consists of separate, coordinated infrastructure changes. After deployment, the infrastructure will ideally remain static until the next change weekend. With AWS, applications now have the option of initiating and automating infrastructure changes either during scheduled deployments or automatically in response to changing user demands on the application. When releasing and deploying applications on AWS, organizations should at least consider how actively the application should be able to participate in the process of ensuring the infrastructure is deployed and configured to best support its business functions.

Conclusion

Many organizations have successfully deployed and operated their cloud applications on AWS. The checklists provided in this whitepaper highlights several best practices that we believe are essential and will help you to increase the likelihood of successful deployments and frustration-free operations. We highly recommend that you leverage these operational and strategic considerations for your existing and new application deployments on AWS.