Revision and Signoff Sheet

Abstract

Strategy-level document specifying CUSTOMER’s approach to deploying applications on Amazon Web Services infrastructure. Documents CUSTOMER’s current approach to architecture, provisioning, and operations on AWS; and also provides the next steps to increase cloud adoption maturity.

Customer Sample– Cloud Playbook

Delivered by:

Professional Services

## Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
|  |  |  |  |
|  |  |  |  |

## Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Position | Date |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Change Record i](#_Toc169879753)

[Reviewers i](#_Toc169879754)

[1 Purpose 3](#_Toc169879755)

[2 Related Documents 3](#_Toc169879756)

[3 Operations Overview 4](#_Toc169879757)

[4 Platform Operations 5](#_Toc169879758)

[5 Provisioning and Service Catalog 12](#_Toc169879759)

[6 Availability Management 13](#_Toc169879760)

[7 System Lifecycle Management 18](#_Toc169879761)

[8 Financial Management 27](#_Toc169879762)

[9 Definition of Terms 29](#_Toc169879763)

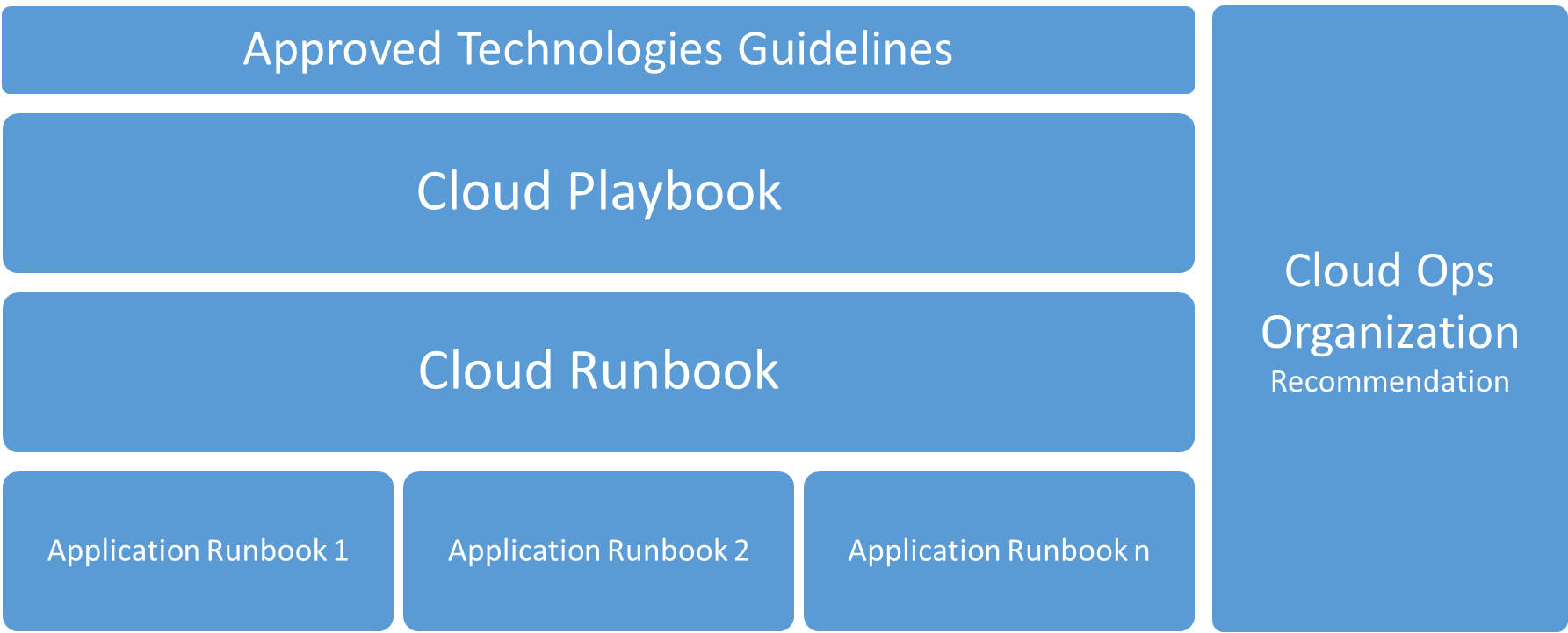
# Purpose

The purpose of this document, referred to herein as “Cloud Playbook” is to be a strategy-level document specifying CUSTOMER’s approach to deploying applications on Amazon Web Services infrastructure. The Cloud Playbook documents CUSTOMER’s current approach to architecture, provisioning, and operations on AWS; and also provides the next steps to increase overall cloud adoption maturity through iterations and incremental improvement.

The intent is for this to be a “living” document that is referred to often and continuously updated as the organization’s experience with AWS increases and the CUSTOMER business functions that surround it mature.

# Related Documents

The Cloud Playbook is part of a series of interrelated documents that address the topics of cloud adoption from technology selections, strategy formation, and cloud execution, all the way through application-specific Runbook details. The level of detail included in these documents is illustrated below, starting with the highest level strategy information at top, pushing down through the most specific and detailed at the bottom.



* Approved Technologies Guidelines represent a framework to guide the organization in selection of technologies to insure an efficient use of resources, improve time to market and the ability to maintain service level operations.
* Cloud Playbook is Strategy-level document specifying CUSTOMER’s approach to deploying applications on Amazon Web Services infrastructure. Documents CUSTOMER’s current approach to architecture, provisioning, and operations on AWS; and also provides the next steps to increase cloud adoption maturity.
* Cloud Runbook is an execution level document that provides the technical detail of how to operate AWS infrastructure.
* Application Runbook(s) provide a view into application specific exceptions from the standard cloud guidelines as outlined in the Cloud Runbook.
* Cloud Ops Organization outlines the resources, skills and teams that are involved in driving the cloud adoption (COE) as well as those who manage the day to day provisioning, usage, monitoring, and cost of cloud resources (Cloud Engineering).

# Operations Overview

As organizations adopt cloud services, the operations organization must transform in order to fully realize the benefits of cloud technologies. Operationalizing your cloud service offering includes adapting the services offered, the manner in which they are offered as well as who and how they are offered. This document is to be a living document that outlines the IT service management features like proactive monitoring, patch, continuity, and financial management, and processes like change and incident management. There are options identified that can be implemented dependent upon the maturity of your organization.

For each category there is a section which identifies the initial strategy that Customer has selected as well as an identified growth opportunity. The Cloud Runbook will expand upon the option Customer has selected and provide prescriptive steps to implement the desired approach.

The specific categories are as follows:

|  |  |
| --- | --- |
| Operations Category | Sub-categories |
| Platform Operations | Tagging  Application Service Level Classification  Application Recovery Classification  Patch Management  Platform Logging |
| Provisioning & Service Catalog | Infrastructure Provisioning  Service Catalog |
| Availability Management | Backup & Recovery (DR)  High Availability |
| Application Lifecycle Management | Application Architecture Patterns  Security Group Patterns  Platform Monitoring  License Management  Network Architecture Patterns  Load Balancers  Firewall Management  Application Logging  Resource Cost Optimization  Configuration Management |
| Financial Management | Chargeback Process  Billing Method  Budgeting and Forecasting  Reporting |

To be optimal, service management and operational functions must be proactive and supported by automation, as opposed to reactive and supported by manual human intervention. This applies both to deploying resources and automating responses to potential issues by designing for failure. Repetitive manual tasks should be reduced through the use of automation to allow operations teams to focus on value-add work.

Operational areas that are important to address early in the cloud journey include Financial Management, Platform Operations, Monitoring & Incident Resolution, Provisioning & Service Catalog, Availability Management and Application Lifecycle Management. Similar to the iterative approach of agile development, there will be an iterative approach to cloud operations.

# Platform Operations

|  |  |  |
| --- | --- | --- |
| Activity | Traditional Approach | Cloud Approach |
| Tagging | Tagging of all resources across an enterprise is not a common feature found in most organizations today. Most organizations utilize a CMDB to track the assets according to cost center or project codes, entering the relevant attribute information in the CMBD manually as the equipment is purchased and then again when it’s provisioned. Usually updates to the CMDB are manual and lag the actual change, especially in scenarios where CMDB’s are used to track the inter-dependencies of applications. | Tags are the building blocks of cloud resource reporting, but they also are functional and are used in everything from auto-scaling to license management. Due to the ephemeral nature of resources in the cloud, it is not necessary to log cloud resources in the same static pattern as physical resources. Instead, a tagging strategy is employed to log resources for tracking, dependency, and chargeback purposes. Additionally, the attributes of the cloud resources being utilized can be readily identified using AWS Config, alleviating the need for manual entry and low-value data entry time.  Key Element to a Cloud Governance Framework   * Cost Transparency * Automation Enablement * Policy Enforcement * Application Correlation * Object Organization   Tags are used by many groups including; Finance, Operations, IRM/Security, Application Owners, and others. Document the Key Fields identified for each report:   * Field Values, Length, Formats * Logical Association of the fields * Typical fields to look for:   + Line Of Business   + Cost Center   + Version   + Owner   + Compliance Domain   + Name   + Environment   + Application   + Tier   Please see the AWS whitepaper which explores various tagging strategy and best practices: <https://d0.awsstatic.com/aws-answers/AWS_Tagging_Strategies.pdf> |
| Application Service Level Classification (SLO/SLA) | Most organizations will define several application service level classifications to communicate the desired uptime of an application (SLO/SLA) of an application. In traditional approaches the SLO/SLA is measured in minutes/days and achieving these metrics is an expensive proposition. | Most customers when moving to the cloud will utilize their existing classification process, with the addition of specifying their classifications with AWS resource tags. In cloud approaches service levels are generally identified in terms of minutes/hours and can be cost effectively architected to achieve these levels.  Patterns:   * Utilize existing classification process without any changes * Ensure AWS services being utilized adhere to your service level classifications * Allows for future improvements with service uptime to grow with evolving business needs |
| Application Recovery Classification (RTO/RPO) | Most organizations will define several application recovery classifications to communicate the desired level of acceptance regarding data loss in the event of a disaster for an application. These are calculated using Recovery Point Objective (RPO) and Recovery Time Objective (RTO). | Most customers when moving to the cloud will utilize their existing classification process, with the addition of identifying their classifications with AWS resource tags.  Patterns:   * Utilize existing classification process without any changes * Ensure AWS services being utilized adhere to your recovery level classifications * Allows for future improvements with recovery times to grow with evolving business needs |
| Patch Management | Most organizations have a standard monthly patch management cadence to maintain or update operating system and off the shelf applications such as database and web hosting software. Critical patches often follow an exception process based on potential business impact.  Typically patching is performed across the enterprise using tools such as Chef, Puppet, and Ansible, Salt, private repos, WSUS, and Microsoft SCCM. | Some organizations take advantage of their existing investment in patch management tools and processes. Often when multiple tools and differing processes are used to address patch management, organizations coalesce around a single enterprise configuration management solution. Below are some of the common options used to address patch management:  Options:   * Some organizations when moving to the cloud will initially utilize their existing patch management process assuming their tool is cloud-aware and/or has the capability to perform the needed functions on the cloud * When using Auto Scaling with EC2, patching can be simplified by updating a single custom Amazon Machine Images (AMI) and re-launching. * As you adopt services such as Amazon RDS and Amazon ElasticBeanstalk you can take advantage of managed platform updates which handle the patch management for these services for you. * As you adopt fully managed services such as Amazon DynamoDB, Amazon SQS, and AWS Lambda, patch management is handled entirely for you. * Many customers today utilize a number of patch management tools and have consolidated around a single solution such as AWS OpsWorks or Chef. |
| Platform Logging | Most organizations do not have a consistent approach or tooling when it pertains to platform logging. However, a pattern of aggregating logs to provide visibility into the enterprise view with various third party tools (ie. Splunk) has begun to emerge. | Organizations building or migrating applications in the cloud have an opportunity to design centralized platform logging as part of the initial architecture.  This brings new insight into the health of a given environment and reduces incident resolution times and drive improvements to the solution.  Below are the options available to support a platform logging solution in the cloud.  Options:   * Amazon CloudWatch is a logging and monitoring service. CloudWatch tracks metrics, monitors log files, sets alarms and reacts to changes in your AWS resources. Amazon CloudWatch is used to gain system-wide visibility into resource utilization, application performance, and operational health * Amazon CloudWatch logs can be consumed by your existing log aggregate tool. * AWS Marketplace and the open source community offer a variety of solutions for centralized logging and analytics |
| Configuration Management | Configuration Management Databases (CMDB) was originally designed as a database to store records of configuration items (CI) throughout their lifecycle as well as their relationships with other configuration items. The CMDB has been traditionally populated with data from static physical environments with long lifecycles. This traditional configuration data is not going to be appropriate for optimized operations in a cloud environment due to the ephemeral nature of cloud services like IP addresses, instance names, and storage volumes. The industry is changing to adapt to capturing not just configuration and relationship information, but also performance information on infrastructure, applications, and services.  Configuration Management Databases will need to be evaluated and structured or modified to reflect data that is really needed to run operations in the cloud. With the dynamic nature of the cloud and capabilities such as auto-scaling and build-test-destroy, Configuration Items may be provisioned and decommissioned within hours. Organizations need to determine what the critical data elements are for each component, application or service; how often they want to update the CMDB; and how to utilize that data. Cloud providers offer multiple ways of configuration management – integration between the provider and client tool, federation of tools, or use of the provider tools. An appropriate tagging strategy and consistent and accurate tagging implementation are critical to identifying and tracking cloud resources.  A CMDB for cloud computing will be even more valuable when micro-services and container architectures are deployed in complex, dynamic, ever-changing cloud configurations. Documenting the relationships among CI’s is paramount when performing any type of compliance assessment, threat impact assessment, or troubleshooting efforts.  In addition, a new type of CI will be seen as more automation is created. Infrastructure as code provides the foundation for IT infrastructure deployment automation and should be captured in industry-standard notation and stored in configuration management tools, just like code. From there the organization can move into continuous integration / continuous deployment (CI/CD) to combine infrastructure and software deployment automation. As code templates are developed and used in the service catalog and to deploy full application stacks, those templates will need to be put through processes such as change control when modified. | Most organizations embrace some of the changes involved in their configuration management process due to the elastic nature of using cloud resources. Review the options below to determine the level of configuration management integration with cloud resources. See Tagging section of this document for more information.  Options:   * Enable AWS Config service to provide dynamic configuration change history and integrate the information into your existing CMDB. This could be a simple pointer to AWS Config   + Ensures the CMBD is always up to date through an automated process.   + Support new capabilities such as resources that automatically are provisioned and de-provisioned with a short life cycle such as Amazon EC2 Auto Scaling. * As your organizational adoption matures, it is possible to build API based integration between AWS and your existing CMDB. A second layer of deeper information that may not be supported or valuable to log in the master CMDB system regarding the cloud infrastructure state, changes, API calls, etc. Instead, this information can be logged in a CMDB-like application hosted by AWS (e.g. AWS Config). Access to this information can be hot linked to the master CMDB and quickly viewed at any time. * Alternatively, it is possible to use AWS Lambda scripts that auto-update their CMDB when resources are deployed or terminated   + If a resource isn't properly tagged, a script can shut it down |

|  |  |
| --- | --- |
| Initial Approach for CUSTOMER (Refer to Runbook for execution details) | |
| Tagging | * CUSTOMER will follow the AWS recommended approach to use tags to identify application service level, cost center, environment type, cost center, organization and more. Specific details of the tagging strategy are described in the Cloud Runbook. * An emphasis on enforcing tagging at launch time to ensure all costs are accounted for and allocated appropriately will be accomplished through required CloudFormation variables and the internal code review.   Growth Path:   * In addition to enforcing tagging at launch, using AWS Lambda (serverless computing) scripts will be used to search and report on resources that are not tagged. |
| Application Service Level | * CUSTOMER will use the existing workflow and process for identifying the Service Class of an application and will also continue to use the Application Recovery Classification using Recovery Point Objective (RPO) and Recovery Time Objective (RTO) as a measure. The currently established RTO/RPO’s are detailed below. * Once the application is identified into a Service Class, that classification will determine the architecture template (see: Application Architecture Patterns section of this document) that best fits the application requirements. * Service Level is directly related to RTO/RPO below. |
| Application Recovery Classification (RTO/RPO) | Current CUSTOMER Recovery Tier Standards:   * Recovery Tier 1 (Mission)   + Time Objective: 24 hours   + Point Objective: ~4 hours * Recovery Tier 2 (Mission)   + Time Objective: Up to 48 hours   + Point Objective: 24 hours * Recovery Tier 3 (Business)   + Time Objective: Up to 72 hours   + Point Objective: 24 hours * Recovery Tier 4 (Business)   + Time Objective: up to 168 hours   + Point Objective: 24 hours * Recovery Tier 5 (Operational)   + Time Objective: 169+ hours   + Point Objective: 24 hours   Growth Path:   * Current RTO/RPO to be evaluated in the future against the cost/benefit of increased availability strategies using AWS. It’s understood that better RTO/RPO’s are attainable a lesser cost on AWS than with traditional architectures with physical data center limitations. Given this, CUSTOMER will take their current Service Class metrics under review in the near future to evaluate whether their RTO/RPO’s are valid given the new capabilities on AWS. |
| Patch Management | * CUSTOMER will explore the use of fully managed (e.g. RDS, DynamoDB, Elastic Beanstalk, etc.) and/or serverless (Lambda) cloud services for every project in an effort to reduce the future need for patching servers on every application. * When patching is required, CUSTOMER delineate between applications that are treated like “pets” vs. “cattle” and will use at least three slightly different approaches for managing and deploying patches to AWS EC2 Compute instances. These approaches will all use scripting and should be repeatable across many instances at one time. These approaches are detailed in the Cloud Runbook.   Growth Path:   * Continued maturity down the path of using more cloud managed service that do not require patching; and when that is not an option, continued maturity in creating application architectures/choosing software vendors that provide the ability to be cloud aware. |
| Platform Logging | * CUSTOMER will use CloudWatch logs with CloudTrail and VPC Flow Logs to review network, security, and system related health and load. Platform Logging will leverage the tagging strategy to report and subsequently notify, regarding resources outside the normal operating ranges.   Growth Path:   * These logs will be sent to the existing CUSTOMER Splunk implementation for further analysis, storage, and or action. |
| Configuration Management | * CUSTOMER will use AWS Config to view real-time AWS resource inventory, configuration history, and enable configuration change notifications. * AWS infrastructure will initially not be stored in the CUSTOMER enterprise CMDB due to the rate of change and the ephemeral nature of the AWS resources; including: instance types, ID’s, IP’s, storage volumes, test/dev environments, etc.   Growth Path:   * This real-time view will enable security and governance of the platform once these individuals are trained and/or are able to consume the outputs of AWS Config into their current systems. An accompanying service, AWS Config Rules will enable CUSTOMER to continuously monitor configuration changes to AWS resources and provide a new dashboard to track compliance status, so an administrator can quickly determine when and how a resource went out of compliance. * CUSTOMER may also explore the case and options for writing Change and Configuration Management details back to their enterprise CMDB |

# Provisioning and Service Catalog

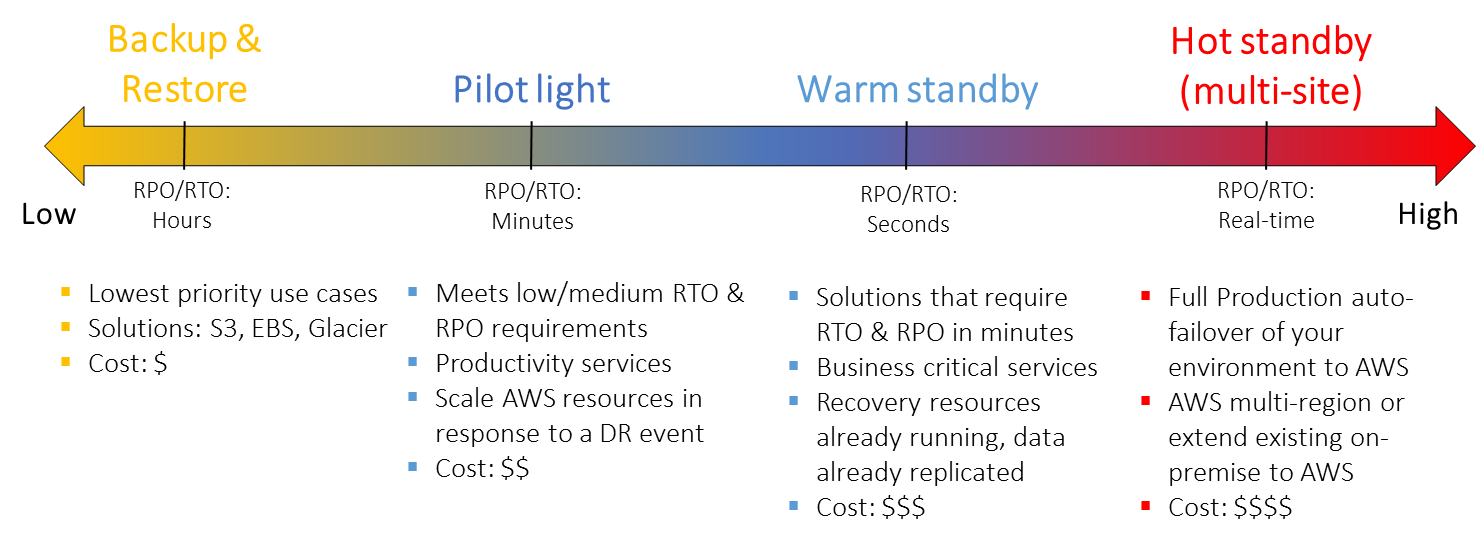
AWS Service Catalog allows organizations to create and manage catalogs of IT services that are approved for use on AWS. Then, end users can quickly discover and deploy commonly used services using a self-service portal.

|  |  |  |
| --- | --- | --- |
| Activity | Traditional Approach | Cloud Approach |
| Provisioning | An application owner documents infrastructure requirements to support an idea or a solution, then submit and ultimately obtaining architecture and design approval. Upon design approval, a request is submitted for financial approval, a purchase order is submitted to the vendors, hardware delivered to the datacenter and then provisioned by a shared services organization.  An additional approach incorporates the scheduling and provisioning of assets with a vendor (MSP) vs on premise data center.  Both of these approaches lead to reduced business agility, slower time to market, significant upfront cost and risk hindering the ability to innovate. | Most organizations initially maintain their existing provisioning process and incorporate either manual or automated provisioning of infrastructure. However, as your adoption extends and matures, organizations select a blend of provisioning and self-service options from the list outlined below:  Options:   * Upon design approval, manually provision infrastructure via the AWS management console or Command Line Interface (CLI). * Upon design approval, automate provisioning of infrastructure by using the AWS CloudFormation service.   + Improved consistency and repeatability while treating your infrastructure as code.   + Enables the ability to version, rollback and manage an application lifecycle as a single unit. * An application owner documents infrastructure requirements through an AWS CloudFormation template file. Then submits and subsequently obtaining architecture and design approval. The next step would incorporate automated provisioning via AWS CloudFormation service as outlined above. * An application owner documents new infrastructure requirements through an AWS CloudFormation template file for architecture and design approval. If the architecture and design is a pattern that would be useful for multiple applications the AWS CloudFormation template file would be included in the AWS Service Catalog.   + For example, 3-tier application running Linux, Apache, Tomcat, and MySQL |
| Service Catalog | Most organizations use an IT service catalog to manage the service lifecycle, including service definition, configuration, continuous improvement and termination. The service catalog is also used to market available services to the business and to provide a means by which IT and its business customers can align around service demand. The service catalog can also be used as a means to manage day-to-day service requests, including online ordering and automated fulfilment. | As organizations mature in their cloud use, there is a general movement toward self service provisioning. To aid in this, some organizations have made approved CloudFormation templates available via their Service Catalog for end user selection.  . |

|  |  |
| --- | --- |
| Initial Approach for CUSTOMER (Refer to Runbook for execution details) | |
| Provisioning | CUSTOMER will utilize the current workflow process within ServiceNow to obtain the initial request for infrastructure and/or services. Then, CUSTOMER Cloud Ops will utilize CloudFormation (infrastructure-as-code) whenever possible to deploy new services. These scripts will be stored in CodeCommit to provide a central repository to the Cloud Engineering team and encourage re-use.  Growth Path:  Expose the AWS Service catalog to enable more self service provisioning to dev and test environments to promote innovation and operational efficiencies. In the early stages of use, the adoption of guardrails around costs, such as including budgets, alarms, will aid in managing the adoption. |
| Service Catalog | Initially all AWS infrastructure will be deployed based on project demand by the Cloud Engineering team using k scripts.  Growth Path:  In the future, CUSTOMER will explore the use of a Service Catalog (ServiceNow, AWS Service Catalog, etc.) to provide other infrastructure consumers inside the organization with the ability to browse a, shopping-like experience for selecting and eventually deploying template solutions in a self-service manner. The future Service Catalog will service two needs:   * + - Browsing and deployment of common application patterns     - Enable self-service prototyping to non-developer or non-DevOps teams   Future decisions and implementation progress of Service Catalog will be documented here. |

# Availability Management

Availability Management encompasses multiple aspects of designing for and operating to insure the availability of applications and services to support the business service level agreement. This includes restoring applications based on Recovery Point and Time objectives. Recovery Point Objectives (RPO) defines how much data loss can be tolerated. While Recovery Time Objective (RTO) defines how quickly the service needs to be restored. Each application should be evaluated based on the business criticality to determine the RTO/RPO requirements. Most organizations classify applications using tiers to convey which recovery option is appropriate for the given application.



Through the use of High Availability (keep your solution running), Backup (make sure your data is safe), and Disaster Recovery (get your data back after a major disaster) the cloud provides comprehensive strategies to meet the desired user experience, SLO/SLA, risk, and cost objectives for each of your applications.

|  |  |  |
| --- | --- | --- |
| Activity | Traditional Approach | Cloud Approach |
| High Availability | Improving application availability utilizing redundant software or infrastructure across multiple data centers. Generally accomplished through the use of cross data center load balancing and synchronous replications. Often establishing an effective HA environment on premise involves a significant investment in time, cost, and resources. | Taking advantage of cloud technologies provides a broad range of HA solutions that meet your organizations availability requirements. Through the evaluation of each of your business applications you can select the appropriate level of availability based upon the options below.  Options:   * Maintain current level of application availability. For example, if an on premise application is in one data center today then provision in one AWS Availability Zone so long as SLA’s meet requirement.   + If selecting this option, policy adherence to snapshots and CloudFormation templates should be utilized to ensure reliable availability. * Improve existing level of application availability by taking advantage of two (2) AWS Availability Zones incorporating Amazon Elastic Load Balancers with traditional databases configured for active/passive synchronous replication. * When using databases that natively offer active/active/active synchronous replication or you intend to consume heavy Amazon EC2 Spot Instances consider using three (3) or more AWS Availability Zones. * For applications requiring high availability globally, consider maintaining duplicate software and infrastructure in multiple AWS Regions and route traffic using Amazon Route 53. |
| Backup & Recovery | A traditional approach to backup and recovery involves a spectrum of hot to cold onsite and offsite duplication of data using various methods dependent upon the criticality of the application. Common patterns for onsite storage include shared file storage using hardware storage appliances, while offsite cold storage is typically tape and disc backup. Backup processes using common methods including full and incremental backups increases complexity when performing restores. Due to the inherent complexity around backup & recovery, many organizations struggle to validate their backup and recovery process. | Most organizations initially maintain their existing backup and recovery processes and software to avoid friction in cloud adoption. However, as your adoption extends and matures, organizations select a blend of backup and recovery options from the list outlined below:  Options:   * Maintain current backup and recovery process and software for Amazon EC2 instances * A hybrid backup and restore process using the same agent based software for EC2 and Amazon RDS snapshots * Implement the use of EBS snapshots combined with a snapshot retention policy * When using Amazon EC2 Auto Scaling consider adopting custom Amazon Machine Images (AMI) over using agent-based or snapshot backups as backups would be redundant. |
| Disaster Recovery | Often disaster recovery requirements take into consideration the following areas:  Redundant power, minimal distance separation relative to geo-location, and duplicate software and infrastructure. This typically leads to significant investment in time, cost and resources to effectively operate and maintain. | Businesses using cloud are more quickly able to recover their IT systems without incurring the expense of an additional physical site. Most organizations classify applications around business criticality and RTO/RPO requirements. For example, 30% of an organizations business critical applications may require recovery in minutes while 70% may require recovery in hours.  Based on application classification, select the disaster recovery option that supports your applications RTO/RPO requirements.  Options:   * Backup & Restore: Non-critical applications with long RTO/RPO requirements in hours. * Pilot Light: Applications with short-to-medium RTO/RPO requirements in minutes. * Warm Standby: Business Critical applications with short RTO/RPO requirements in seconds. * Hot Standby with Multi-site: Business Critical applications with extremely short RTO/RPO requirements with automatic failover in near real time. |

|  |  |
| --- | --- |
| Initial Approach for CUSTOMER (Refer to Runbook for execution details) | |
| High Availability | * For new or refactor projects, CUSTOMER will develop applications to the work with multiple AZ’s and ELBs. This will create applications which are less coupled and able to support application failover from one AZ to another, or the loss of a complete AZ.  |  |  | | --- | --- | | **Application Recovery Classification (RTO/RPO)** | **AWS Recovery Pattern** | | Recovery Tier 1 | Hot Standby (multi-site) | | Recovery Tier 2 | Pilot-light or Warm Standby | | Recovery Tier 3 | Backup and Restore | | Recovery Tier 4 | Backup and Restore | | Recovery Tier 5 | Backup and Restore |   Growth Path:   * To decrease downtime and improve the manageability of applications across the RTO/RPO spectrum CUSTOMER will evaluate the use of “Blue/Green deployment” techniques. This strategy reduces the risk of bang-bang cut-overs and can also be used to evaluate new feature/functions in Production. |
| Backup/  Recovery | CUSTOMER will tag applications with recovery tiers as a mechanism to identify a schedule for snapshots and retention to be applied. If the need for a restore occurs, CUSTOMER will restore from the snap shot and AMI for system recovery. Refer to High Availability matrix identified above  Growth Path:  To be determined by CUSTOMER |
| Disaster Recovery | Patterns such as Pilot Light, Warm Standby, Hot Standby and Multi site availability are possibilities dependent upon business need. Refer to High Availability matrix identified above  Growth Path:  To be determined by CUSTOMER |

# System Lifecycle Management

|  |  |  |
| --- | --- | --- |
| Activity | Traditional Approach | Cloud Approach |
| Application Architecture  Patterns | In traditional solution development, application architectures are determined on a per project basis. Each project will assume the overhead associated with application architecture review and approval. Generally this process can add a significant amount of time and human capital to a project. | For new applications, AWS customers have been discovering cloud specific IT architecture patterns, driving even more efficiency, scalability, and reuse. Identifying approved reusable patterns for the architecture reduces the time spent on each project and allows more time to address the business solution the application is intended to support. For example:   * Public (DMZ) Patterns   + 1-Tier Application Architecture   + 2-Tier Application Architecture   + 3-Tier Application Architecture * Internal Patterns   + 1-Tier Application Architecture   + 2-Tier Application Architecture   + 3-Tier Application Architecture |
| Security Group Architecture | Most organizations have a set of hardware, software or a combination of both that secure the network at the perimeter from unauthorized access in or out. | A security group acts as a virtual firewall for your instance, cloud service and/or API level to control inbound and outbound traffic.  Similar to application architecture patterns, security groups also have a set of reusable patterns which can be standardized in an organization to improve efficiencies in rolling out business solutions. For example:   * Use 1 security group per application tier * Use 1 security group for "inter-region" one-to-many situations, such as Jumpbox * Use 1 security group for "inter-region" many-to-one situations, such as instance-based monitoring agents |
| Event Monitoring | Typical approach utilizes agents installed on all servers/VM’s that report back to a centralized service to aggregate all monitoring events. These monitoring events are then reviewed by Operations Engineers and if necessary, converted to an incident.  Some of the challenges associated with this approach is the difficulty in maintaining the latest version of the agent as well as the overall cost of the monitoring solution. | Most organizations initially maintain their existing event monitoring solution and incident management processes to avoid friction in cloud adoption. However, as your adoption extends and matures, organizations select a blend of monitoring and incident management options from the list outlined below:  Options:   * Maintain current monitoring solution and incident management processes for Amazon EC2 instances * A hybrid of existing monitoring solutions and AWS monitoring service e.g. Amazon CloudWatch, as well as AWS Partner monitoring solutions.   + When using Amazon EC2 Auto Scaling consider a monitoring solution that is aware of scale out and scale in events.   + Most AWS services outside of Amazon EC2 require a cloud aware event monitoring solution. * Use Amazon CloudWatch for all event monitoring to gain system-wide visibility into resource utilization, application performance, and operational health using a combination of built-in metrics as well as custom metrics generated by your application. |
| Incident Monitoring | Most organizations utilize an incident management process that encompasses the recording of, acting on, communicating progress of, and providing notification of, active incidents. | Operating cloud solutions incurs minimal change relative to current incident management procedures. The initial reporting of the incident follows the same process you utilize today, however, an additional escalation path to AWS would be incorporated to help troubleshoot incidents that utilize AWS services. Below are the AWS Support Plans to select from based upon your organizations planned use (align this to your multi-account pattern).  Options:   * Experimenting with AWS   + Individual developers exploring the potential of AWS, looking for access to technical support resources to help quickly and effectively get started.   + Recommended plan: Developer Support * Production use of AWS   + Businesses looking for guidance and best practices to enable availability, scalability, and security of production workloads -reducing the need for reactive support   + Recommended Plan: Business Support * Business Critical use of AWS   + Businesses whose success is directly linked with the performance of workloads and applications, benefiting from high-touch, proactive/preventive services.   Recommended Plan: Enterprise Support |
| License Management | Most organizations manage licensing at an enterprise, portfolio and application level. Licensing at the portfolio level is used to maintain centralized governance of usage and pricing.  The process typically includes a step of preapproved software that is maintained to reduce specific application software licensing approval.  Enforcement of licensing is commonly managed through a combination of manual and automated methods including yearly audits, centralized licensing servers, etc. | Most organizations chose a hybrid of licensing models for easing some of the challenges involved in maintaining license compliance through options such as AWS license included. Consider some of the following options for inclusion in your license management process  Options:   * Maintain the existing process for managing portfolio licenses and application software exceptions. * Incorporate AWS License-included for supported operating systems and Amazon RDS databases. * Incorporate AWS Bring Your Own License for supported operating systems and databases. * Incorporate AWS Marketplace applications which typically utilizes AWS license included. * Tagging can be utilized to aid in the tracking of licenses to aid in the management |
| Network Architecture Patterns | Most organizations have cross functional teams which to design the network architecture dependent upon each applications needs. Many times this portion of a project involves a gated review and members from security, network, firewall, storage and application teams. | Most organizations adopt a network architecture which allows for delineation between internet accessibility as well as the use of multiple Availability Zones to aid in high availability.  Options:  VPC  Start with 2 subnets per Availability Zone   * Use subnets to delineate Internet accessibility (public/private); avoid creating subnets for specific applications or application tiers (web-tier, app-tier, data-tier)   + Consider larger subnets over smaller ones (/24 or larger)   VPC Connectivity   * Maintain the use of your current internet connection * Utilize your existing VPN unless bandwidth requirements exceed your current capacity * If you exceed VPN bandwidth, consider the use of AWS Direct Connect |
| Load Balancer | Most organizations have an approval process to make changes to rules and pool membership across centrally managed load balancers.  Often organizations have challenges around maintaining consistent rule and pool memberships across multiple load balancers as they are manually entered. Additionally, on premise hardware load balancers can be cost prohibitive and are often only deployed for Production and Test application environments. | Organizations moving to the cloud are exposed to a significant mental change around using one or two load balancers per application. Due to the shift of load balancers becoming more application level constructs, load balancers are typically configured by application development teams.  Below are the options available to support a load balancer solution in the cloud.  Options:   * Most organizations, when adopting the cloud embrace the concept of application specific load balancers often used in front of web and app tiers of an application.   + AWS offers Amazon Elastic Load Balancers (ELB) to provide TCP, HTTP/S, and SSL off loading * Organizations desiring a single pane of glass to manage rules and pool membership, may choose to use a load balancer solution from the AWS Marketplace or open source community. |
| Firewall Management | Most organizations have an approval process to make changes to inbound/outbound rules across centrally managed firewalls.  Often organizations have challenges around maintaining consistent rule sets across multiple firewalls as they are manually entered. Additionally, firewalls are deployed at a minimum as a perimeter security posture leaving most internal applications exposed to lateral VM to VM attacks. | Organizations moving to the cloud are exposed to a significant mental change around perimeter firewalls becoming micro zone application tier firewalls. Due to the shift of firewalls becoming more application level constructs, firewalls are typically configured by application development teams and enforced through automated centralized policy  Below are the options available to support a firewall solution in the cloud.  Options:   * Most organizations, when adopting the cloud embrace the concept of micro zoning which requires firewalls around each application tier that comprises an application.   + AWS offers mandatory firewalls which are referred to as Security Groups. All AWS Services have zero trust by default.   + AWS also offers Network Access Control Lists, which are optional firewalls that restrict subnets, but should only be used as required by compliance. * Organizations desiring a single pane of glass to manage firewall rules, may choose to use a firewall solution from the AWS Marketplace or open source community. |
| Application Logging | Most organizations do not have a consistent approach when it pertains to application logging. A fraction of applications within an organization utilize a custom application logging solution that is often decentralized and designed specifically for that application. | Organizations building new applications in the cloud have an opportunity to design centralized application logging as part of the initial application architecture.  This brings new insight into the health of an application that can reduce incident resolution times and drive improvements to the application.  Below are the options available to support an application logging solution in the cloud.  Options:   * New applications being designed for the cloud often utilize centralized logging.   + AWS offers Amazon CloudWatch logs to provide centralized logging and analytics   + AWS Marketplace and the open source community offer a variety of solutions for centralized logging and analytics * Existing applications that have been migrated to the cloud can also take advantage of centralized logging but may require minor refactoring. |

|  |  |
| --- | --- |
| Initial Approach for CUSTOMER (Refer to Runbook for execution details) | |
| Application Architecture Pattern | CUSTOMER has chosen to create, use and enforce standard architecture practices wherever possible to decrease one-off solutions. This increases the simplicity of training Cloud Engineering staff. The details, including graphical representations of these architectures are included in the Cloud Runbook.   * CUSTOMER Application Architecture Patterns: * Public (DMZ) Patterns   + 1-Tier Application Architecture   + 2-Tier Application Architecture   + 3-Tier Application Architecture * Internal Patterns   + 1-Tier Application Architecture   + 2-Tier Application Architecture   + 3-Tier Application Architecture   Growth Path:  The building blocks of these architectures can contain Amazon Machine Images (AMI)’s to further automate and speed the provisioning time of new resources. There are choices to be made as to how to manage the lifecycle of AMI’s, ranging from “fully-baked AMI’s” to “just-enough Operating System (jeOS)”. These strategies are detailed in the Runbook. |
| Security Groups | CUSTOMER has chosen to create, use and enforce standard security groups wherever possible to decrease one-off solutions. This increases the simplicity of training Cloud Engineering staff. The details, including graphical representations of these groups and how they apply to common architectures are included in the Cloud Runbook. In general, CUSTOMER will follow:  Use 1 security group per application tier   * Use 1 security group for "inter-region" one-to-many situations, e.g. Jumpbox * Use 1 security group for "inter-region" many-to-one situations, e.g. instance-based monitoring agents * May incorporate exceptions for PCI data |
| Event Monitoring | CUSTOMER to maintain the use of SolarWinds for performance monitoring where possible. Will evaluate on an app-by-app basis for now. Where not possible to use only SolarWinds to monitor all aspects of an application in the cloud, AWS CloudWatch will be used either in conjunction or as a replacement to SolarWinds.   * Note: AWS CloudWatch not only performs monitoring of your application and AWS infrastructure, but is seamlessly integrated and essential to key cloud features like AutoScaling. Not using CloudWatch may limit the ability to use such features.   Growth Path:  Future consideration will evaluate use of CloudWatch as a full replacement to SolarWinds. CloudWatch collects and tracks standard and customer metrics, collects and monitors log files, provides alarms, and integrates with other AWS services to automatically react to changes in your AWS resources. |
| Incident Monitoring | Strategy to be defined by Customer |
| License Management | CUSTOMER will bring their own licenses (BYOL) to AWS in situations where CUSTOMER has an Enterprise Agreement (EA) with the license provider and that provider allows for license portability to AWS. CUSTOMER will tag any and all resources with those licenses deployed in AWS, so license count/auditing can be performed periodically.  Growth Path:  Use “license-included” or open-source powered managed cloud services to reduce CUSTOMER licensing cost (including reducing licensing count for future Enterprise Agreements.) |
| Network Architecture | During the AWS Platform Jumpstart engagement with AWS ProServe, CUSTOMER decided on the following Network, VPC, AZ, and Connectivity decisions.   * Approved Regions   + US-EAST- 1 (Virginia)   + US-WEST-2 (Oregon)   + EU-WEST (Ireland)   + AP-NORTHEAST-1 (Tokyo) * Number of Availability Zones   + Minimum and maximum of 2 Availability Zones per Region * Number of VPC’s   + 3 Amazon VPCs in Prod AWS account (Prod, Management and Workspaces)   + 3 Amazon VPC in Non-Prod AWS account (Prod, Management and Workspaces)   + 1 Amazon VPC in Lab/POC AWS account   + Any number of Amazon VPCs in Fuse AWS account   + Any number of Amazon VPCs for Pivotal Cloud Foundry   + 4 Amazon VPCs for Dev, Stage, QA, and Prod in Customer @ Home AWS account * VPC Connectivity   + Management VPC will be peered to all VPCs in the same region   + VPN connections initially, being the process of instantiating Direct Connect lines   Growth Path:  Establish 4 AWS Direct Connect lines in 4 AWS Regions   * Dublin Ohio Data Center   + 1 - 10Gbs line and port speed to US-EAST-1   + 1 - 10Gbs line and port speed to US-WEST-2   + 1 - 30Mbs line and 1 Gbs port speed to EU-WEST   + 1 - 30Mbs line and 1 Gbs port speed to AP-NORTHEAST-1 * McGaw Illinois Data Center   + 1 - 10Gbs line and port speed to US-EAST-1   + 1 - 10Gbs line and port speed to US-WEST-2   + 1 - 30Mbs line and 1 Gbs port speed to EU-WEST   + 1 - 30Mbs line and 1 Gbs port speed to AP-NORTHEAST-1 |
| Load Balancer | CUSTOMER decision is to use Amazon Elastic Load Balancers (ELB) in front of each internal or external stateless end-point to avoid hard coding IP addresses and to increase the reliability and resiliency of the service / server. |
| Firewall Management | Strategy to be defined by Customer   * Decision points to be weighed – The continued use of on-premise architectures in AWS (e.g. Palo Alto, perimeter security, IDS/IPS, etc.) vs. cloud-native tools and network/application-level security groups. |
| Application Logging | CUSTOMER will use Amazon CloudWatch logs to provide application monitoring and logging as well as use advanced features such as Amazon EC2 AutoScaling. |

# Financial Management

|  |  |  |
| --- | --- | --- |
| Activity | Traditional Approach | Cloud Approach |
| Chargeback Process | All costs are centralized in traditional IT accounting. One central department pays for all IT equipment and activities, typically out of the CTO or CIO’s budget, and these costs are treated as corporate overhead shared evenly by multiple departments.  In an IT chargeback accounting model, individual cost centers are charged for their IT service based on use and activity. As a result, all IT costs are “zeroed out” because they have all been assigned to user groups. IT is no longer considered overhead, instead it can be viewed as part of each department’s business and operating expenses (OpEx).  Typically, allocation methods are based on # of users, # of transactions etc. | Most organizations require greater visibility into their cloud costs and have the ability to provide a granular view into the resources being utilized in the cloud, by environment, business unit, application owner and cost center. This enables IT a means to provide a chargeback model based upon actual resource consumption. |
| Billing Method | Most organizations will have centralized billing for IT resources which are then allocated back to the business units based upon and agreed upon measure. | Most organizations will utilize the Consolidated Billing feature to consolidate payment for multiple Amazon Web Services (AWS) accounts by designating one of them to be the payer account. This enables your organization to see a combined view of AWS charges incurred by all accounts, as well as get a cost report for each individual account associated with your payer account. |
| Budgeting and Forecasting | Most organizations follow a budgeting cadence for capital and operating expenses. Once procured there is generally a monthly review of actual to plan expenditures and subsequent project budgets are adjusted. | To aid in budget management and forecasting, most organizations will use a combination of alerts on CloudWatch as well as AWS Trusted Advisor to manage the spending run rate relative to the operational plan.  Options:   * Maintain existing controls to manage and forecast operational expense * Monitor your AWS costs by utilizing CloudWatch to create billing alerts that notify your organization when the usage of AWS services exceeds thresholds that were defined by your organization. * Utilize Cost Explorer to view graphs of your spend data for up to the last 13 months, and forecast how much you are likely to spend for the next three months. You can use Cost Explorer to see patterns in how much you spend on AWS resources over time, identify areas that need further inquiry, and see trends that you can use to understand your costs |
| Reporting |  | Most organization review billing reports to gain information about their usage of AWS resources and estimated costs for that usage. AWS can generate billing reports that break down your estimated costs in different ways:   * By the hour, day, or month * By each account in your organization * By product or product resource * By organizationally defined tags |

|  |  |
| --- | --- |
| Initial Approach for CUSTOMER (Refer to Runbook for execution details) | |
| Chargeback Processing | CUSTOMER currently uses an aggregate chargeback model at the org level (Med/Pharma). These costs are budgeted yearly and a fixed charge is allocated throughout the year to the Org. With the move toward cloud adoption, CUSTOMER has the ability to charge at a more granular level based upon cost center consumption. However, timing differences with the CUSTOMER planning process and cloud adoption can cause disparity on individual business unit financial reports. Therefore, the following process was discussed as a compromise around adoption and chargeback:   * Fiscal year 1 – Hybrid Chargeback/Showback model – all resources will include a tag at the org level allowing the chargeback model to remain consistent and individual business unit financials to remain in line with fiscal year plans. However, monthly, a report will be generated to provide visibility into consumption/costs by business unit (showback). This data will then be utilized during the planning process to budget for a consumption based chargeback model. * Fiscal year 2 – Chargeback Model – individual business units will receive a monthly charge based upon consumption. This allows for greater visibility as well as opportunities for the business units to control/manage their IT spend. |
| Billing Method | CUSTOMER will utilize the Consolidated Billing feature of AWS to combine all CUSTOMER accounts into one payer account. This provides the ability to see a combined view of charges incurred by all AWS accounts as well as get a cost report for each individual account associated with the payer account. |
| Budgeting and Forecasting | An AWS CloudWatch billing alert will be setup via the Account Billing Console and visible via CloudWatch alerts. Additionally, to ensure proper tagging, a Lambda script will be executed to provide a list of instances which are not properly tagged for follow up. |
| Reporting | Strategy to be defined by Customer |

# Definition of Terms

The following table contains the definitions of terms used in this document.

| Term | Definition / Description |
| --- | --- |
| **AWS** | Amazon Web Services |
| **Availability Zone** | http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-regions-availability-zones |
| **AWS Region** | http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-regions-availability-zonee |
| **CI** | Configuration Item |
| **Cloud HSM** | http://docs.aws.amazon.com/cloudhsm/latest/gsg/cloud-hsm-overview.html |
| **CMS** | Configuration Management System |
| **CMDB** | Configuration Management Database |
| **Customer** | Defined as a business entity or group who procure the service. Usually {Customer} Application Teams or Business Units. |
| **DHCP** | Dynamic Host Configuration Protocol |
| **Direct Connect** | <http://aws.amazon.com/directconnect/> |
| **DNS** | Domain Name Service |
| **DR** | Disaster Recovery |
| **EBS** | <http://aws.amazon.com/ebs> |
| **EBS Encryption** | <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html>  <http://media.amazonwebservices.com/AWS_Securing_Data_at_Rest_with_Encryption.pdf> |
| **EC2** | <http://aws.amazon.com/ec2> |
| **ELB** | AWS Elastic Load-Balancer (<http://aws.amazon.com/elasticloadbalancing/>) |
| **End User** | Defined as an individual user of a system. Differs from a Customer. |
| **Enhancement** | Refers to a request for a change that is not related to restoring existing service, but has focus on an improvement or other change to existing services. |
| **Excusable Downtime** | Refers of the Scheduled Uptime, the aggregate number of hours in a month that a defined ‘system’ is unavailable due to action or inaction by The Customer, The Customers Agents, or due to a Force Majeure event, or is unavailable at the request of The Customer. |
| **Glacier** | aws.amazon.com/glacier |
| **IAM** | AWS Identity and Access Management (IAM) – <http://aws.amazon.com/iam/>  AWS Multi-Factor Authentication – <http://aws.amazon.com/mfa/>  IAM Roles – <http://docs.aws.amazon.com/IAM/latest/UserGuide/WorkingWithRoles.html>  IAM Roles for Amazon EC2 – <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html>  AWS Security Token Service – <http://docs.aws.amazon.com/STS/latest/UsingSTS/Welcome.html>  IAM Best Practices & Use Cases - <http://docs.aws.amazon.com/IAM/latest/UserGuide/IAMBestPracticesAndUseCases.html> |
| **Incident** | Any event that is not part of the standard operation of a service and that causes, or may cause, an interruption to, or a reduction in, the quality of that service. Refers to an occurrence where Technology Infrastructure support is required. Typically, an incident is raised by an End User, or through monitoring activities |
| **Internet Gateway** | <http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html> |
| **IP** | Internet Protocol |
| **ITSM** | IT Service Management |
| **LAN** | Local Area Network |
| **NTP** | Network Time Protocol (http://www.ntp.org/) |
| **Problem** | A [Problem](javascript:void(0);) is a condition often identified because of multiple [Incident](javascript:void(0);)s that exhibit common symptoms. Problems can also be identified from a single significant Incident, indicative of a single error, for which the cause is unknown, but for which the impact is significant. |
| **RFC** | Request For Change |
| **RBAC** | Role Based Access Control |
| **RDS** | Relational Database Service (an AWS Managed Database offering)  <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_BestPractices.html> |
| **Redshift** | <http://docs.aws.amazon.com/redshift/latest/dg/best-practices.html>  <http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html> |
| **RPO** | Recovery Point Objective |
| **RTO** | Recovery Time Objective |
| **S3** | Amazon Simple Storage Service (aws.amazon.com/s3)  <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html> |
| **Service Request** | Every Incident not being a failure in the [IT](javascript:void(0);) Infrastructure. Any event that does not result in failure of the IT Infrastructure, but is a requirement to enable/enhance existing or a new service |
| **SLA** | Service Level Agreement |
| **SLM** | Service Level Management |
| **SPOF** | Single Point Of Failure |
| **Squid Proxy** | [www.squid-cache.org/](http://www.squid-cache.org/)  aws.amazon.com/articles/5995712515781075 |
| **STS** | AWS Security Token Service - <http://docs.aws.amazon.com/STS/latest/APIReference/Welcome.html> |
| **System** | Defined as an application, technology infrastructure component or a service that separately defined for the purposes of this Blueprint. Examples include; LAN, WAN, Phone System, Technical Support etc. |
| **Tags** | <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html> |
| **VPC** | AWS Virtual Private Cloud (<http://aws.amazon.com/vpc/)> |
| **Virtual Private Gateway** | <http://docs.aws.amazon.com/AmazonVPC/latest/NetworkAdminGuide/Introduction.html> |
| **VPN** | Virtual Private Network |
| **WAN** | Wide Area Network |