Migrating AWS Resources to a New Region

March 2013

Simon Elisha, James Bromberger & Peter Stanski

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

Table of Contents

Abstract

This document is intended for experienced customers of Amazon Web Services who would like to migrate existing resources to a new AWS region. Customers may wish to migrate for a variety of reasons. In particular, if a new region has been made available closer to their user base, customers may wish to locate various services geographically closer to those users.

This document is not intended to be a "step-by-step" or "definitive" guide, rather, it provides customers with a variety of options and methods to migrate various services that they may require in a new region.

Introduction

Amazon Web Services now operates in a number of regions worldwide serving customers in over 190 countries. For a number of AWS services, you can choose which geographic region where you would like that service to be delivered. Regions are dispersed and located in separate geographic areas (United States, Europe, Asia Pacific, South America, etc.) and have multiple Availability Zones. By using separate Availability Zones, you can further protect your applications from the failure of a single location. By using separate AWS regions, you can design your application to be closer to specific customers and achieve lower latency and higher throughput. AWS had designed the regions to be isolated from each other so that you can achieve greater fault tolerance and improved stability in your applications.

Scope of AWS Resources

Whilst most AWS services operate in a region-independent fashion, the following services operate across all regions and require no migration:

- AWS Identity and Access Management (IAM)
- AWS Management Console
- Amazon CloudWatch

Further, as all services are accessible using API endpoints, you do not necessarily need to migrate all components of your architecture to the new region depending on your application. For example, you may migrate Amazon Elastic Compute Cloud (Amazon EC2) instances, but retain existing Amazon Simple Storage Service (Amazon S3) and Amazon CloudFront configurations.

An updated list of available AWS products and services by a specific region is available on AWS Website – Global Infrastructure Section[1].

Before you migrate to a new region, we recommend that you check to make sure that AWS products and services are readily available.

[1] http://aws.amazon.com/about-aws/globalinfrastructure/regional-product-services

AWS IAM and Security Considerations

AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users.

IAM users are created and managed within the scope of an AWS account, rather than a particular region—no migration of users or groups is required.

When migrating to a new region, it is important to note any defined policy restrictions on IAM users. For example, Amazon Resource Names (ARNs) might restrict you to a specific region. For more information, please refer to Identifiers for IAM Entities[2] section in the IAM User Guide.

IAM is a core security service that enables customers to add specific policies to control how a user can access AWS resources. Some policies may affect time of day access (which may require consideration due to time-zone differences), use of new originating IP addresses, whether SSL connections need to be used, how users have been authenticated, and whether multi-factor authentication (MFA) devices should be used.

Because AWS Identity and Access Management (IAM) underpins security, we recommend that you undertake a careful review of your security configuration policies, procedures, and practices before a region migration.

Compute & Networking

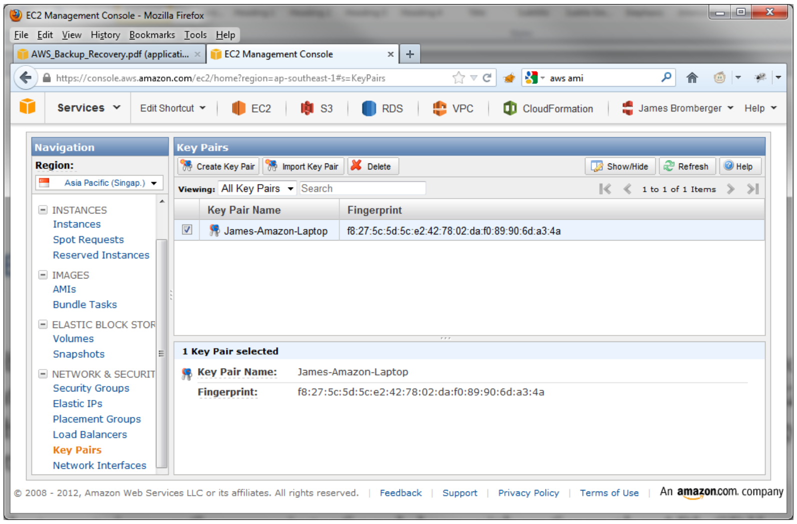
Migrating Amazon EC2 Instances

Amazon EC2 is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Migrating an instance is a case of copying the data and images, ensuring that the security groups and SSH keys are present, and then restarting fresh instances.

SSH Keys

Amazon Web Services does not keep any user SSH private keys after they have been generated by our customers. These public keys are made available to Amazon EC2 instances when they are running (under Linux operating systems, these normally are copied into the relevant user's `~/.ssh/authorized_keys` file).

[2] http://docs.amazonwebservices.com/IAM/latest/UserGuide/Using_Identifiers.html#Identifiers_ARNs



	Key Pair Name	Fingerprint
☑	James-Amazon-Laptop	f8:27:5c:5d:5c:e2:42:78:02:da:f0:89:90:6d:a3:4a

1 Key Pair selected

Key Pair Name: James-Amazon-Laptop

Fingerprint: f8:27:5c:5d:5c:e2:42:78:02:da:f0:89:90:6d:a3:4a

Figure 1 – Key pairs in the AWS Management Console

Users are able to retrieve a fingerprint of each key from the API, SDK, command line, or the console.

SSH public keys are only stored per region; AWS does not copy or synchronize the configured customer SSH keys between regions. It is up to customers to determine if they will use separate SSH keys per region, or the same SSH keys in several regions. **Note**: You can log onto an existing Linux instance in the source region, obtain a copy of the public key (from `~/.ssh/authorized_keys`), and import this into the destination region.

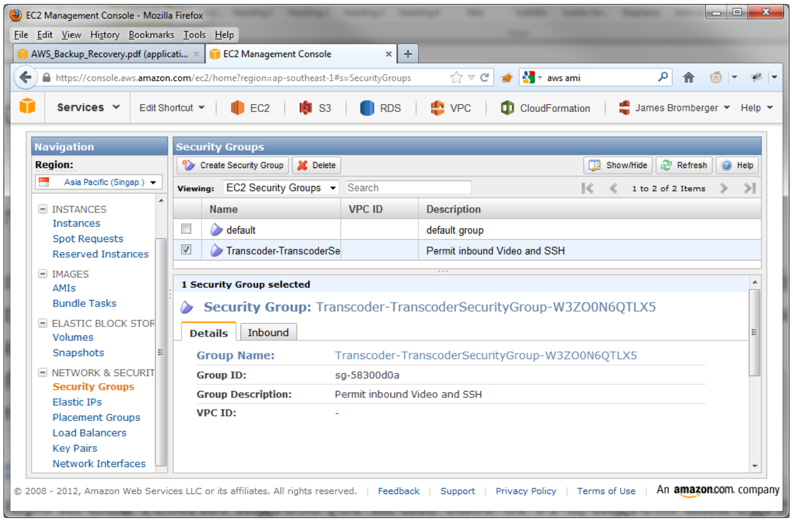
It is important to note that Auto Scaling launch configurations and AWS CloudFormation templates may refer to SSH keys using the key pair name. In this case, the user must take care to either update any Auto Scaling launch configuration or AWS CloudFormation template to use keys that are available in a new region, or deploy the public key with the same key pair name to the new region.

For more information, see About AWS Security Credentials[3].

Security Groups

Security groups in Amazon EC2 restrict ingress (or in the case of VPC, ingress and egress) traffic to a group of Amazon EC2 instances. Each rule in a security group may refer to the source (or in VPC, the destination) by either a CIDR notation IPv4 address range (*a.b.c.d/x*), or using the security group identifier (*sg-XXXXXXXX*).

[3] http://docs.amazonwebservices.com/AWSSecurityCredentials/1.0/AboutAWSCredentials.html.



	Name	VPC ID	Description
□	default		default group
☑	Transcoder-TranscoderSe		Permit inbound Video and SSH

1 Security Group selected

Security Group: Transcoder-TranscoderSecurityGroup-W3ZO0N6QTLX5

Group Name:	Transcoder-TranscoderSecurityGroup-W3ZO0N6QTLX5
Group ID:	sg-58300d0a
Group Description:	Permit inbound Video and SSH
VPC ID:	-

Figure 2 – Security group configuration in the AWS Management Console

Each security group can only exist within the scope of one region. The same name may exist in multiple regions, but have different definitions of what traffic is permitted to pass.

Every instance being launched must be a member of a security group. If a host is being started as part of an Auto Scaling launch configuration or an AWS CloudFormation template, then the required security group must exist (AWS CloudFormation templates may often define the security group to be created as part of the template).

It is vital that you review your configured security groups to ensure that the required level of network access restrictions is in place.

To export a copy of the definitions of existing security groups (using the command line tools), use the following command:

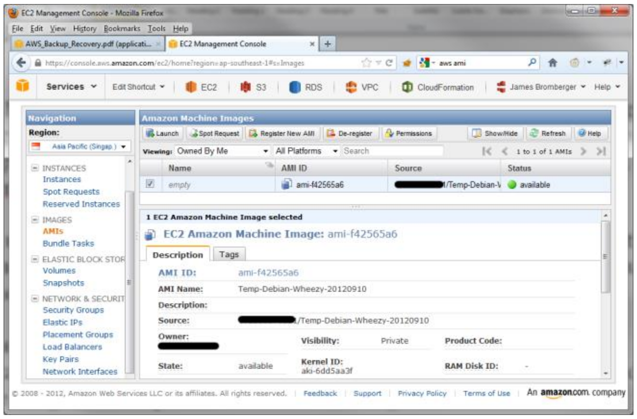
```
ec2-describe-group -H --region <sourceregionname> > security_groups.txt
```

For more information, see the Security Groups section of the Amazon EC2 User Guide[4]..

Amazon Machine Images

An Amazon Machine Image (AMI) is a special type of pre-configured operating system image used to create a virtual machine (an instance) within the Amazon EC2 environment. Each AMI is assigned an identifier, of the form “ami-*XXXXXXXX*”, where ‘X’ is a hexadecimal value (0-9, A-F).

[4] http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/using-network-security.html



Name	AMI ID	Source	Status
☑ empty	ami-f42565a6	[redacted]/Temp-Debian-V	● available

1 EC2 Amazon Machine Image selected

EC2 Amazon Machine Image: ami-f42565a6

AMI ID: ami-f42565a6

AMI Name: Temp-Debian-Wheezy-20120910

Description:

Source: [redacted]/Temp-Debian-Wheezy-20120910

Owner: [redacted] Visibility: Private Product Code:

State: available Kernel ID: aki-6dd5aa3f RAM Disk ID: -

Figure 3 – AMIs in the AWS Management Console

Each AMI is unique per region and AMIs do not span multiple regions. However, the same content of an AMI may be available in other regions (for example, Amazon Linux 2012.09 or Windows Server 2008 R2) and each region will have its own unique AMI ID for its copy of this data.

You are able to create your own AMIs from running instances, and use these as a starting point for launching additional instances. These user-created AMIs are assigned a unique AMI ID within the region.

AMI IDs are used within Auto Scaling launch configuration and AWS CloudFormation templates. If you plan to use Auto Scaling or AWS CloudFormation, then you need to update the AMI ID references to match the ones that exist in the target region.

Migration of AMIs across regions is supported using the EC2 AMI Copy function[5]. EC2 AMI copy enables you to copy an AMI to as many regions as you like from the AWS Management Console, the Amazon EC2 CLI, or the Amazon EC2 API. EC2 AMI copy is available for Amazon EBS-backed AMIs as well as instance-store-backed AMIs, and is operating-system-agnostic.

Each copy of an AMI results in a new AMI with its own unique AMI ID. Any changes made to the source AMI during or after a copy are not propagated to the new AMI as part of the AMI copy process; you must recopy the AMI to the destination regions to copy the changes made to the source AMI.

Note: Permissions and user-defined tags applied to the source AMI are not copied to the new AMIs as part of the AMI copy process. After the copy is complete, you may apply any permissions and user-defined tags to the new AMIs.

Amazon Elastic Block Store Volumes

Amazon Elastic Block Store (Amazon EBS) is a block storage volume that can be presented to an instance. An Amazon EBS volume can be formatted with a specific file system type such as NTFS, vFAT, ext4, xfs, etc. Amazon EBS volumes may contain the operating system boot volume or be used as an additional data drive (Windows), or mount point (Linux).

[5] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/CopyingAMIs.html

Amazon EBS volumes can be migrated using the cross-region Amazon EBS snapshot copy capability[6]. This enables snapshots of Amazon EBS volumes to be copies between regions using either the Console, API call or Command Line.

Key capabilities of this feature include:

- The AWS Management Console shows the progress of a Snapshot copy in progress, you can check the percentage complete there.
- You can initiate multiple Snapshot Copy commands simultaneously either by selecting and copying multiple Snapshots to the same region, or by copying a snapshot to multiple regions in parallel. The in-progress copies do not affect the performance of the associated Amazon EBS Volumes.
- The console-based interface is push-based; you log in to the source Region and tell the console where you'd like the Snapshot to end up. The API and the command line are, by contrast, pull-based and you must run them within the destination Region.

The entire process takes place without the need to use external tools or to perform any configuration.

High-level migration process:

- Identify relevant Amazon EBS volumes to migrate (you may choose to make use of tagging to assist in identification)
- Identify which volumes can be copied with the application running, and which require an outage (shutdown of the application) Amazon EBS Snapshot copy accesses a snapshopt of the primary volume, rather than the volume itself – so the application may need to be shutdown during the copy process to ensure the latest data is copied across.
- Create the necessary Amazon EBS snapshots and wait for them to be in “Complete” status.
- Initiate the Amazon EBS snapshot copy using either the console, API or CLI.
- Create new Amazon EBS volumes at the destination region by selecting the relevant snapshots and using the “create volume from snapshot” functionality.

Volumes and Snapshots

Amazon EBS volumes can currently be from 1 GB and up to 1 TB in size in 1 GB increments, and can be used with disk management tools (such as Logical Volume Manager (LVM) or Windows Disk Manager) to span/stripe across multiple block devices. You can stripe multiple Amazon EBS volumes together to deliver higher performance instances to applications.

Volumes that are in constant use may benefit from having a snapshot taken, especially if there are multiple volumes being used in RAID 1 stripe or part of an LVM volume group.

Provisioned IOPS volumes are another way to increase Amazon EBS performance. These volumes were designed to deliver predictable, high performance for I/O intensive workloads such as databases. To enable Amazon EC2 instances to fully use the IOPS provisioned on an Amazon EBS volume, you can launch selected Amazon EC2 instance types as

[6] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-copy-snapshot.html

“EBS-Optimized” instances. Before a region migration, we recommend that you check that Availability Zones in the target region support such instances.

For more information, see Increasing EBS Performance in the Amazon EC2 user Guide[7]..

Elastic IP Addresses

Elastic IP addresses are assigned to an account from the pool of addresses for a given region. As such, an Elastic IP address cannot be migrated between regions.

The recommendation is to update your time-to-live (TTL) value on your DNS that points to this Elastic IP address, and reduce it to an amount that is a tolerable delay in DNS cache expire, such as 300 seconds (5 minutes) or less. Any decrease in DNS TTL could result in an increase in DNS requests, increase load on your current DNS service, and affect charges from your DNS service provider.

DNS changes can be done more optimally by taking a staged approach to TTL modifications. As an example:

- Current TTL for www.example.com (which points to an Elastic IP address) is 86400 (1 day)
- Modify the TTL for www.example.com to 300 seconds (5 minutes), and schedule work for 2 days’ time
- Monitor the increase of DNS traffic in this period
- At the start of the day of the scheduled work, reduce the TTL for www.example.com further; later, optionally reduce the TTL further depending upon load on your DNS infrastructure (possibly 10 seconds)
- 10 minutes after the last change, update the A record to point to a new Elastic IP address in the new region
- After a short period, confirm that traffic is being adequately serviced, and then increase the TTL back to 5 minutes (300 seconds)
- After another period of operation, return the TTL to normal level.

Elastic Load Balancing

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances.

Elastic Load Balancing cannot be migrated to a new region. Instead, a new Elastic Load Balancing service needs to be launched in the target region, containing a new set of Amazon EC2 instances spanning desired Availability Zones within the service group.

Before a region migration, we recommend that you review the source and target region Availability Zones to confirm that matching levels of zones exist. In scenarios where extra Availability Zones are discovered, application load balancing and scalability could need to be revised. This could lead to further assessments of CloudWatch alarms and thresholds that are used for Auto Scaling group configuration.

[7] http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/EBSPerformance.html

Furthermore, associated SSL certificates on the old Elastic Load Balancing service need to be added to the new Elastic Load Balancing service, and health check conditions added to verify Amazon EC2 instance health tests.

Launch Configurations and Auto Scaling Groups

Auto Scaling allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define.

Use the following commands to capture the current Auto Scaling and launch configuration definitions. Note that Auto Scaling is not currently configurable using the AWS Management Console; however, other third-party management tools support it:

```
as-describe-auto-scaling-groups –H --region <sourceregionname> >
autoscale_groups.txt
as-describe-launch-configs –H --region <sourceregionname> >
launch_configs.txt
```

These extracted Auto Scaling groups and launch configuration settings will reference AMIs, security groups, SSH key pairs as they exist in the source region - see the above sections on migrating these to the target Region, and then create new Auto Scaling groups and launch configurations in the target region using new AMI IDs and security groups.

For more information on Auto Scaling groups and launch configurations, see Basic Scenario in Auto Scaling in the Auto Scaling User Guide.[8]

Reserved Instances Considerations

Many customers take advantage of greatly reduced pricing of Reserved Instances for Amazon EC2, Amazon Relational Database Service (Amazon RDS), and Amazon ElasticCache. Reserved Instances (or reserved cache nodes) are assigned to a specific instance type (size) within a set Availability Zone in a specific region for a period of one or three years.

Reserved instances are available in three utilization levels: Light, Medium, and Heavy. The upfront cost and per-hour charges vary between these utilization levels, as well as between different geographic regions.

Light and Medium utilization Reserved Instances also are billed by the instance-hour for the time that instances are in a running state. If an instance does not run, there is a zero usage charge. Partial instance-hours consumed are billed as full hours. Heavy utilization Reserved Instances offer the maximum savings over On Demand Instances and are billed for every hour during the entire term (which means that customers are charged the hourly fee regardless of whether any usage has occurred during an hour).

If you have purchased Reserved Instances and wish to migrate them to a different region, we recommend that you first sell them in the Reserved Instances Marketplace. As soon as they are sold, the billing switches to the new buyer and you are no longer being billed for the Reserved Instances. The buyer will then continue to pay the remainder of the term. In order to get savings over On-Demand Instances, you can either buy Reserved Instances for a shorter term in the region where you are migrating from the Reserved Instance Marketplace or purchase directly from AWS. Reserved Instance

[8] http://docs.amazonwebservices.com/AutoScaling/latest/DeveloperGuide/US_BasicSetup.html

Marketplace makes it easy to "migrate" your billing to a new region. For more details information on how to buy and sell Reserved Instances, please see How to Purchase Reserved Instances and the Reserved Instance Marketplace[9] pages.

We recommend that you carefully assess cost implications prior to purchasing and selling Reserved Instances before undertaking a migration to a new region.

Migrating Amazon Virtual Private Clouds

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a private, isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define.

Amazon Virtual Private Clouds (Amazon VPC) exist per region, but may span multiple Availability Zones within that region. They cannot be moved or migrated to a new region. However, you can create a new VPC in a new target region, and could potentially use the same IP address ranges as the existing VPC.

You should consider the IP address ranges being used so that routing continues to work during a region migration. It is possible to create a duplicate VPC in the new region using the same internal IP address ranges, so long as management at the customer gateway is done correctly and overlapping ranges are never accessed simultaneously.

Migrating AWS Direct Connect Links

AWS Direct Connect is a service that links physical infrastructure to the AWS services. One or more fiber connections are provisioned in an AWS Direct Connect location facility.

If you want to provision new links in a new region, you must request a new AWS Direct Connect service and provision any tail circuits to their infrastructure. Note that charges for AWS DirectConnect vary per geographic location.

Existing connections can be terminated at any time when no longer required.

AWS has relationships with a number of different peering partners at each geographic region. You can find an updated list of AWS Direct Connect Amazon Partners that can assist with service provisioning at http://aws.amazon.com/directconnect/partners/.

Using Amazon Route 53 to Aid the Migration Process

Amazon Route 53 is a highly available DNS service that is available from all AWS regions and edge locations worldwide. DNS can be very effective when managing a migration scenario, as it can assist in gracefully migrating traffic from one location to another by routing traffic either by single cutover, or gradually. By adding new DNS records for the copy of an application in the destination region, you can a) test access to the application and b) choose when to cut over to the new site/region.

One approach is to use Weighted Resource Record Sets functionality; this enables customers to determine a percentage of traffic that should be routed to each particular address when using the same DNS name. For example, to route all traffic to the existing region and none to the new region, the following configuration could be used:

```
www.mysite.com  CNAME elbname.sourceregion.com      100
www.mysite.com  CNAME elbname.destinationregion.com 0
```

[9] http://aws.amazon.com/ec2/reserved-instances/marketplace/

When it is time to perform the migration, the weighting on these records are flipped as follows:

```
www.mysite.com  CNAME elbname.sourceregion.com      0
www.mysite.com  CNAME elbname.destinationregion.com 100
```

This causes all new DNS requests to resolve to the new region. **Note**: Some clients may continue to use the old address if they have cached their DNS resolution, if a long TTL exists, or if a TTL update has not been honored.

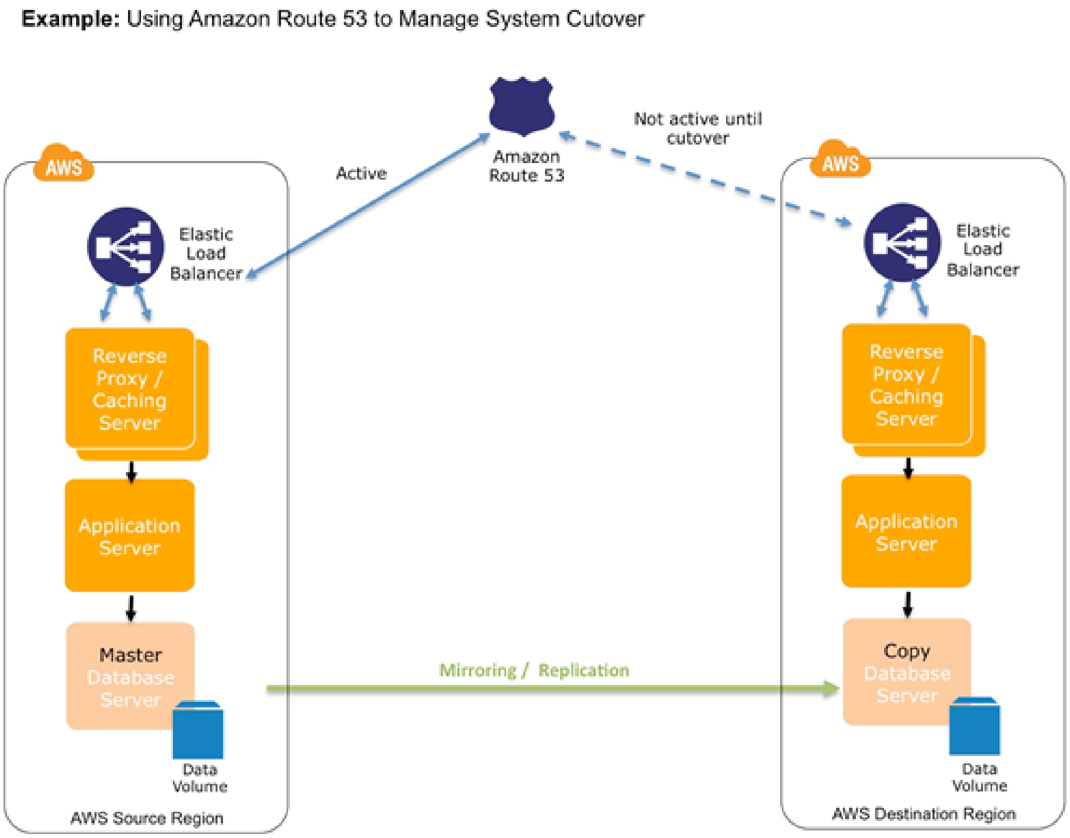


Figure 5 – Using Amazon Route 53 to facilitate region migration

It is also possible to perform gradual cutover using varied weightings as long as the application supports a dual region operational model. For more information, see Creating Weighted Record Sets in the Amazon Route 53 User Guide[10].

[10] http://docs.amazonwebservices.com/Route53/latest/DeveloperGuide/WeightedResourceRecordSets.html

Storage & CDN

Migrating Amazon Simple Storage Service Buckets

Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web.

When an Amazon Simple Storage Service (Amazon S3) bucket is created, it resides physically within a single AWS region. Access could be affected by network latency when it is accessed from another remote region. You should pay careful attention to any references to the Amazon S3 buckets and their geographic distribution as this may introduce latency.

To migrate an Amazon S3 bucket, you need to create a new Amazon S3 bucket in the new target region, and then copy the data to it. The new bucket requires a universally unique name.

For more information on Amazon S3 bucket naming rules, see Bucket Restrictions and Limitations in the Amazon S3 User Guide. [11]

Virtual Hosting with Amazon S3 Buckets

You might be hosting web sites through the static website hosting feature of Amazon S3. For more information, see Web Site Hosting in the Amazon S3 User Guide[12].

For simplicity and user friendliness, customers often use a DNS CNAME alias for their Amazon S3 hosted web content, from an URL like http://bucketname.s3.amazonaws.com to http://my.bucketname.com. Through a CNAME alias, the specific Amazon S3 URL endpoint is abstracted from the web browser. For more information, see Virtual Hosting in the Amazon S3 User Guide[13].

When an Amazon S3 bucket previously used as a static web site has been migrated to a new AWS region (under a new bucket name), it can continue being accessed using the same user friendly name by changing the CNAME alias within the DNS record (for example, Amazon Route 53) from the old bucket name to the new.

Move Objects By Using the AWS Management Console

The AWS Management Console has the ability to copy or move multiple objects between Amazon S3 buckets. By manually selecting one or more objects and selecting **Cut** or **Copy** from the pop-up menu, these items can then be pasted or moved into a target Amazon S3 bucket. The destination bucket may reside in the same region or another geographic region.

[11] http://docs.amazonwebservices.com/AmazonS3/latest/dev/BucketRestrictions.html
[12] http://docs.amazonwebservices.com/AmazonS3/latest/dev/WebsiteHosting.html
[13] http://docs.amazonwebservices.com/AmazonS3/latest/dev/VirtualHosting.html

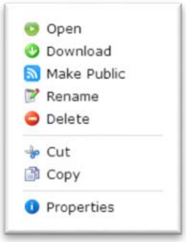


Figure 4 – Setting permissions on an Amazon S3 object using the AWS Management Console

Moving Objects By Using Third Party Tools

To copy or move Amazon S3 objects between buckets a variety of third party tools are available. You can check for AWS Partner products at the AWS Technology Partner page, using the Storage search refinement[14].

Copying Using the API/SDK

Programmatic copy/move operations of Amazon S3 objects between buckets can be done through the Amazon SDK APIs. For more information on Amazon S3 object level operations, see Operations on Objects in the Amazon S3 API Reference[15].

To speed up the object copying process, the use of PUT Object - Copy operation can be used. A PUT Object - Copy operation performs a GET and then a PUT API call in a single operation, which can copy to a destination bucket. For more information, see PUT Object – Copy in the Amazon S3 API Reference[16].

Migrating Amazon CloudFront Distributions

Amazon CloudFront is a content delivery service that operates from the numerous AWS edge locations worldwide (35 at this time). AWS CloudFront delivers customer data in configuration sets called a distribution. Each distribution has one (or more in the case of cache behaviors) configured origins.

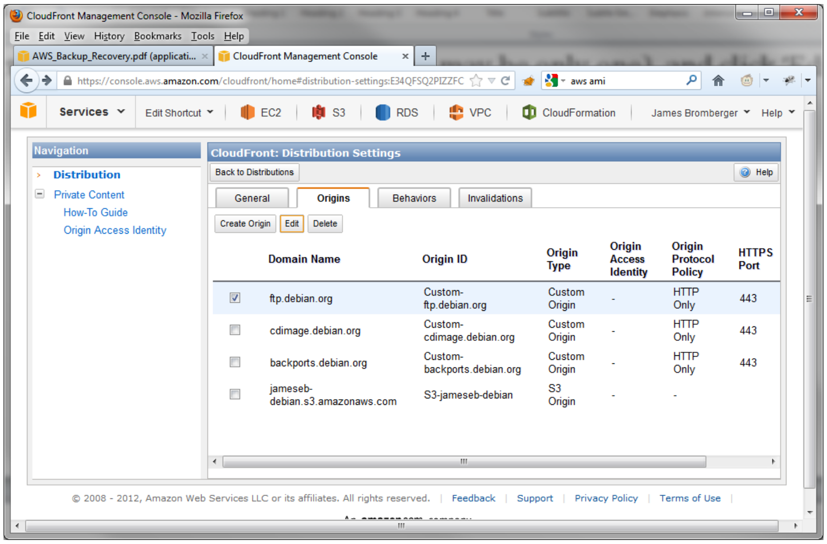
Each origin may be an Amazon S3 bucket, or a web server, including web servers running from within Amazon EC2 (in any AWS region worldwide).

To update an origin in the console

1. Move your origin server or bucket to the new region by referring to the relevant section of this document for Amazon EC2 instances or Amazon S3 buckets.
2. In the Amazon CloudFront console, select the distribution, and click **Distribution Settings**.
3. Select the **Origins** tab.

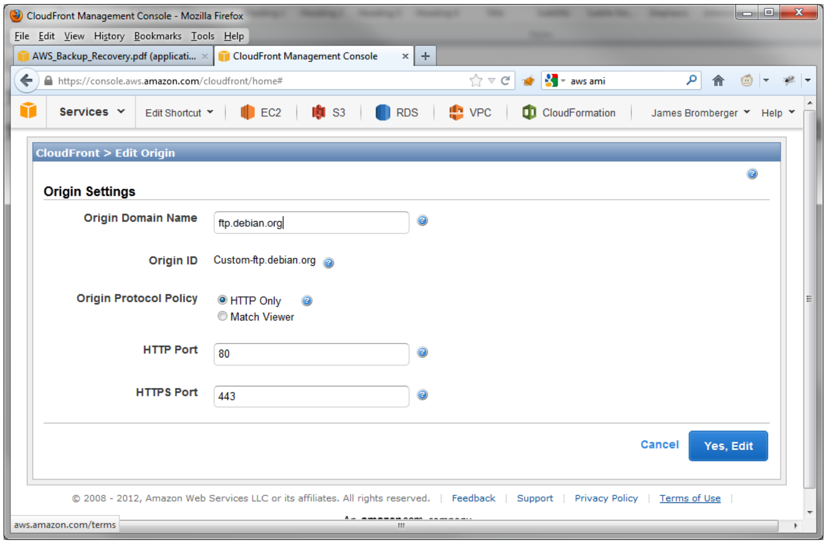
[14] https://aws.amazon.com/solution-providers?business_software_id=6&selection=business_software_id&type=isv
[15] http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTObjectOps.html
[16] http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTObjectCOPY.html

4. Select the origin to edit (there may be only one), and click **Edit**.


CloudFront Management Console - Mozilla Firefox
File Edit View History Bookmarks Tools Help
AWS_Backup_Recovery.pdf (applicati...
CloudFront Management Console
https://console.aws.amazon.com/cloudfront/home#distribution-settings:E34QFSQ2PIZZFC
aws ami
Services
Edit Shortcut
EC2
S3
RDS
VPC
CloudFormation
James Bromberger
Help
Navigation
Distribution
Private Content
How-To Guide
Origin Access Identity
CloudFront: Distribution Settings
Back to Distributions
Help
General
Origins
Behaviors
Invalidations
Create Origin
Edit
Delete
© 2008 - 2012, Amazon Web Services LLC or its affiliates. All rights reserved.
Feedback
Support
Privacy Policy
Terms of Use

	Domain Name	Origin ID	Origin Type	Origin Access Identity	Origin Protocol Policy	HTTPS Port
☑	ftp.debian.org	Custom-ftp.debian.org	Custom Origin	-	HTTP Only	443
□	cdimage.debian.org	Custom-cdimage.debian.org	Custom Origin	-	HTTP Only	443
□	backports.debian.org	Custom-backports.debian.org	Custom Origin	-	HTTP Only	443
□	jameseb-debian.s3.amazonaws.com	S3-jameseb-debian	S3 Origin	-	-	

- Update **Origin Domain Name** with the new server or bucket name.


CloudFront Management Console - Mozilla Firefox
File Edit View History Bookmarks Tools Help
AWS_Backup_Recovery.pdf (applicati...
CloudFront Management Console
https://console.aws.amazon.com/cloudfront/home#
aws ami
Services
Edit Shortcut
EC2
S3
RDS
VPC
CloudFormation
James Bromberger
Help
CloudFront > Edit Origin
Cancel
Yes, Edit
© 2008 - 2012, Amazon Web Services LLC or its affiliates. All rights reserved.
Feedback
Support
Privacy Policy
Terms of Use
aws.amazon.com/terms

Origin Settings

Origin Domain Name: ftp.debian.org

Origin ID: Custom-ftp.debian.org

Origin Protocol Policy: ● HTTP Only ○ Match Viewer

HTTP Port: 80

HTTPS Port: 443

- Click **Yes, Edit**.

For more information, see Listing, Viewing and Updating CloudFront Distributions in the CloudFront Developer Guide[17].

[17] http://docs.amazonwebservices.com/AmazonCloudFront/latest/DeveloperGuide/HowToUpdateDistribution.html

Migrating Amazon Glacier Storage

Amazon Glacier is the AWS deep archive storage service. It is designed to handle large volumes of data that are infrequently accessed.

Amazon Glacier currently offers customers a free tier of data retrieval: 5% of the total storage in Amazon Glacier, pro-rated per month, which is around 0.17% per day; retrieval of more than this amount of data incurs additional charges.

Data stored in Amazon Glacier is often quite large, and you might want to use the AWS Import/Export service to transfer data between regions instead of online transfers.

Strategy 1: Online transfer

This strategy requires that each archive is less than 5 TB in size; while Amazon Glacier supports individual archives of up to 40 TB, Amazon S3 has an object size limit of 5 TB.

To transfer Amazon Glacier storage online

1. Calculate the complete size of your data in Amazon Glacier, and find out the size that is free to retrieve per day (Size * 0.17%).
2. Find an archive in your Amazon Glacier vaults that matches this size closely, and schedule it to be retrieved in Amazon Glacier to the Amazon Glacier staging area.
3. When this archive is available, copy it to a temporary Amazon S3 bucket, optionally using the reduced redundancy storage (RRS) option. This is because the copy process to the target region may exceed 24 hours, and retrieved Amazon Glacier objects are only made available for 24 hours in the staging area.
4. Using a micro instance (to keep costs low) in the target region with enough local (Amazon EBS) storage, download the object (the archive file).
5. Add this file into Amazon Glacier in the target region.
6. Delete the temporary copy of the archive file on the Amazon EBS volume, from the temporary Amazon S3 bucket in the source region, and from Amazon Glacier in the source region.

Strategy 2: Offline transfer

This strategy requires a customer-supplied storage device (removable disk) for transferring between regions; the size of this disk needs to be equal to or larger than the size of the archive being transferred.

To transfer Amazon Glacier storage offline

- Schedule an AWS Import/Export job for the data archive to be transferred to offline storage (disk) at the source region. Identify the disks being sent for export with the job details.
- Wait to receive the exported archive disks.
- Schedule an AWS Import/Export job in the target region.
- Identify the disks being sent for AWS Import/Export service with the job details and send to the target region.

Before an Amazon Glacier region migration, we recommend that you confirm that the target and source regions support the AWS Import/Export service.

Database

Migrating Amazon RDS Services

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business.

Database Security Groups

Amazon RDS has its own set of security groups that restrict access to the database service, using either a CIDR notation IPv4 network address, or an Amazon EC2 security group. Each Amazon RDS security group has a name, and exists only in one AWS region (just as an Amazon EC2 security group does).

Database Instances and Data

Customers may find that they need to schedule downtime in an application to quiesce the data, move the database, and resume operation. The high-level overview is:

- Stop all transactions or take a snapshot (however, changes after this point in time are 'lost' and might need to be re-applied to the target Amazon RDS instance).
- Using a temporary Amazon EC2 instance, dump all data from Amazon RDS to a file.
- For MySQL, make use of the mysqldump tool. You might want to compress this dump (see bzip or gzip).
- For MS SQL, make use of SQL Server Management Studio or the SQL Database Migration wizard. **Note**: Amazon RDS does not support Microsoft SQL Server backup file restores.
- For Oracle, make use of the Oracle Export/Import utility or the Data Pump feature (see http://aws.amazon.com/articles/Amazon-RDS/4173109646282306).
- Copy this data to an instance in the target region using standard tools such as CP, FTP, or Rsync.
- Start a new Amazon RDS instance in the target region, using the new Amazon RDS security group.
- Import the saved data.
- Verify that the database is active and has your data present.
- Delete the old Amazon RDS instance in the source region.

For more information on importing data into the Amazon RDS database service, see Importing Data into a DB Instance in the Amazon RDS User Guide.[18]

[18] http://docs.amazonwebservices.com/AmazonRDS/latest/UserGuide/ImportData.html

Migrating Amazon DynamoDB

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.

While DynamoDB replicates data between Availability Zones in the same region, it does not natively replicate data between regions. However, you can use Amazon Elastic MapReduce (EMR) to perform this function. **Note**: The DynamoDB tables being referenced must exist in both the source and destination regions.

For more information on using Amazon EMR, see Amazon EMR for DynamoDB in the Amazon DynamoDB User Guide[19].. For more information on specific export/import instructions, see EMR Hive Commands in the Amazon DynamoDB User Guide[20].

A third-party solution is also available from http://www.cloudally.com/, which can manage the backup/restoration for you and can restore to an alternate region.

When you use any third party solution, we recommend that you share only specifically secured IAM user credentials that are deleted after the migration takes place.

Migrating Amazon SimpleDB

Amazon SimpleDB is a highly available and flexible non-relational data store that offloads the work of database administration. Developers simply store and query data items via web services requests and Amazon SimpleDB does the rest.

To copy Amazon SimpleDB data between AWS regions, you need to create a specific job or script that extracts the data from the Amazon SimpleDB domain in one region and copies it to the relevant destination in another region. This job should be hosted on an Amazon EC2 instance and would use the relevant SDK that suits your purposes and expertise.

Migration approaches include:

- Establishing simultaneous connections to the new and old domain, querying the existing domain for data, and then putting data into the new domain.
- Extracting data from the existing domain and storing it in a file (or set of files) and then putting that data into the new domain. We recommend that you use the API call BatchPutAttribute to increase performance and decrease the number of API calls performed.

A third-party solution is also available from http://backupsdb.com/, which may suit your needs.

When you use any third party solution, we recommend that you share only specifically secured IAM user credentials that are deleted after the migration takes place.

[19] http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/EMRforDynamoDB.html
[20] http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/EMR_Hive_Commands.html

Migrating Amazon ElastiCache

Amazon ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud.

At this time, the content of Amazon ElastiCache cannot be enumerated without knowing separately all the keys that are present within it.

The recommended approach is to start a new Amazon ElasticCache cluster, and let it start to populate itself through application usage.

Application Services

Migrating Amazon Simple Queue Service Queues

Amazon Simple Queue Service (Amazon SQS) offers a reliable, highly scalable, hosted queue for storing messages as they travel between computers.

Amazon SQS queues exist per region. To migrate the data in a queue, you need to drain the queue from the source region and insert into a new queue in the target region.

When migrating a queue, it is important to note if you need to continue to process the messages approximately in order.

When order is not important:

1. Create a new queue in the target region.
2. Configure applications that enqueue messages to write to the new queue in the target region.
3. Configure applications that read messages form the Amazon SQS queue to read from the new queue in the target region.
4. Have a script that repeatedly reads from the old queue, and submits to the new queue.
5. When the old queue in the source region is empty, delete it.

When order is important:

- Create a new queue in the target region.
- Create an additional new temporary queue in the target region.
- Configure applications that enqueue messages to write to the new queue in the target region.
- Configure applications that read messages form the SQS queue to read from the new temporary queue in the target region.
- Have a script that repeatedly reads from the old queue, and submits to the new temporary queue.
- When the old queue in the source region is empty, delete it.
- When the temporary queue is empty, reconfigure applications to read from the new queue, and delete the temporary queue.

Migrating Amazon Simple Notification Service Topics

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

Amazon SNS topics exist per region. These can be re-created in a target region manually through the AWS Management Console, or programmatically using command-line tools or direct API calls.

To list the current Amazon SNS topic in a designated region, use the following command:

```
sns-list-topics --region <sourceregionname>
```

For more information on the Amazon Simple Notification Service command-line interface tools, see http://aws.amazon.com/developertools/3688.

Deployment & Management

Migrating with AWS CloudFormation

AWS CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.

You can use AWS CloudFormation sample templates or create your own templates to describe the AWS resources, and any associated dependencies or runtime parameters required to run applications. For more information, see http://docs.amazonwebservices.com/AWSCloudFormation/latest/UserGuide/Welcome.html.

While many customers use AWS CloudFormation for the creation of development, test, and multiple production environments within a single AWS region, these same templates can be reused in other regions. You can address disaster recovery and region migration scenarios by running such a template with minor modifications in another region.

Commonly, AWS CloudFormation templates can be readily used by changing the mapping declarations to substitute region-specific information, such as the unique IDs for AMIs, which vary across regions as shown below:

```
"Mappings" : {
  "RegionMap" : {
    "us-east-1"      : { "AMI" : "ami-97ed27fe" },
    "us-west-1"      : { "AMI" : "ami-59c39c1c" },
    "us-west-2"      : { "AMI" : "ami-9e901dae" },
    "eu-west-1"      : { "AMI" : "ami-87cef2f3" },
    "ap-southeast-1" : { "AMI" : "ami-c44e0b96" },
    "ap-northeast-1" : { "AMI" : "ami-688a3d69" },
    "sa-east-1"      : { "AMI" : "ami-4e37e853" }
  }
}
```

For more information on mapping declarations, see Mapping Declaration in the AWS CloudFormation User Guide

Capturing Environments by Using CloudFormer

CloudFormer is a prototype tool that enables customers to create AWS CloudFormation templates from the pre-existing AWS resources.

You provision and configure application resources using your existing processes and tools. After these resources have been provisioned within your environment within an AWS region, the CloudFormer tool takes a "snapshot" of the resource configurations. These resources are placed into an AWS CloudFormation template, enabling you to launch copies of the application environment through the AWS CloudFormation console.

The CloudFormer tool is intended to create a starting point for an AWS CloudFormation template so that you can customize the template further. For example, you can:

- Add parameters to enable stacks to be configured at launch time.
- Add mappings to allow the template to be customized to the specific environments and geographic regions.
- Replace static values with the Ref and Fn::GetAtt functions to flow property data between resources where the value of one property is dependent on the value of a property from a different resource.
- Fill in your Amazon EC2 instance user data to pass parameters to Amazon EC2 instances at launch time.
- Customize Amazon RDS database instance names and master passwords.

For more information on setting up CloudFormer to capture a customer resource stack, see http://www.youtube.com/watch?v=KIpWnVLeP8k.

Note: CloudFormer does not currently support the importation of VPC-related configuration.

API Implications

When programmatic access is required for connecting to AWS regions, publically defined endpoints must be used for API service requests. While some web services allow you to use a general endpoint that does not specify a region, these generic endpoints do resolve to the service's specific regional endpoint.

For more information on the authoritative list of current regions and service endpoint URLs, see http://docs.amazonwebservices.com/general/latest/gr/rande.html.

Updating Customer Scripts and Programs

You may need to update your self-developed scripts and programs that interact with the AWS API (either directly, or using one of the SDKs or command-line tools to ensure that they are communicating with the appropriate regional endpoint.

Each SDK has its own format for specifying the region being accessed. The command-line tools generally support the `–region` parameter.

Important Considerations

Do not leave your AWS certificate or private key on the disk. Clear out the shell history file in case you typed secret information in commands or in environment variables.

Do not leave any password active on accounts. Make sure that the image does not include the public SSH key in the `authorized_keys` files. This leaves a back door into other people's servers even if they do not intend to use it.

It is good practice to use the options [`--region]`, `[--kernel`], `[--ramdisk`] explicitly whenever applicable even though those options are optional.

Verify whether any IP address associations are associated with the AMI. If so, remove them or modify them with the correct details post migration.

Closing Remarks

When you undertake any type of system migration, we recommend comprehensive planning and testing. You should be sure to plan all elements of the migration, with fail-back processes for unanticipated outcomes. AWS makes this process easier by enabling cost-effective testing and the ability to retain the existing system infrastructure until the migration is successfully completed.

Document Revisions

November 2012

- Initial release

December 2012

- Added EBS Snapshot copy (inter-region copy)
- Indicate pending AMI copy
- Remove legacy AMI copy

January 2013

- Removed outdated constraint on S3 support for DNS Zone Apex

March 2013

- Added details of AMI Copy capability