AWS - Cloud Migration 6R's

Reference:

https://aws.amazon.com/cloud-migration/

Check out this cheat sheet on the 6 R's of migration as well as other AWS migration services:

https://tutorialsdojo.com/aws-cheat-sheet-aws-migration-strategies-the-6-rs/

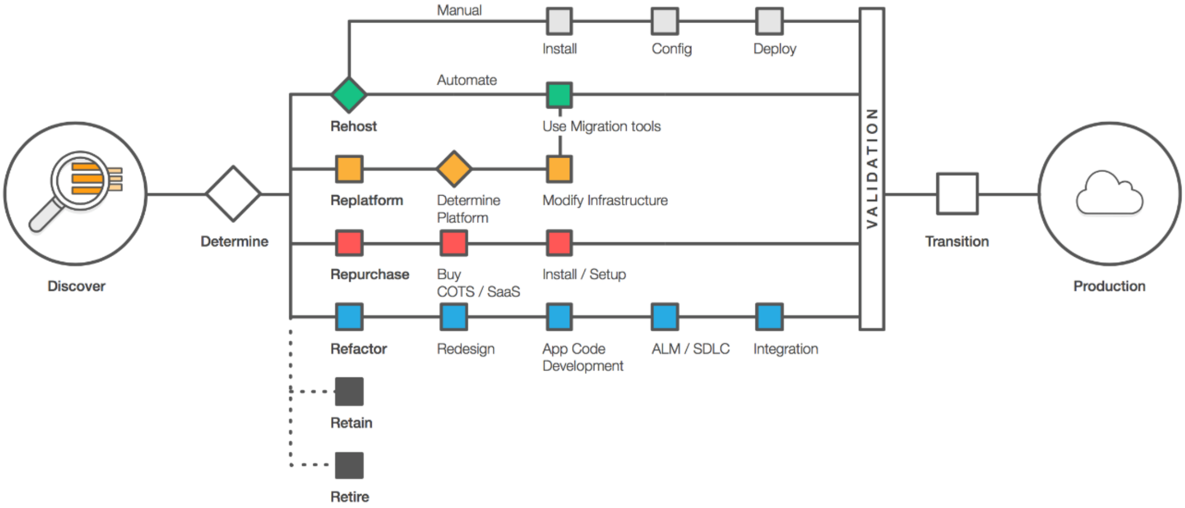
Organizations usually begin to think about how they will migrate an application during Phase 2 (*Portfolio Discovery and Planning*) of the migration process.

This is when you determine what is in your environment and the migration strategy for each application.

The six approaches detailed below are common migration strategies employed and build upon "The 5 R's" that Gartner Inc, a global research and advisory firm, outlined in 2011.

You should gain a thorough understanding of which migration strategy will be best suited for certain portions of your portfolio.

It is also important to consider that while one of the six strategies may be best for migrating certain applications in a given portfolio, another strategy might work better for moving different applications in the same portfolio.



1. Rehost ("lift and shift") - In a large legacy migration scenario where the organization is looking to quickly implement its migration and scale to meet a business case, we find that the majority of applications are rehosted. Most rehosting can be automated with tools such as AWS SMS although you may prefer to do this manually as you learn how to apply your legacy systems to the cloud.

You may also find that applications are easier to re-architect once they are already running in the cloud. This happens partly because your organization will have developed better skills to do so and partly because the hard part - migrating the application, data, and traffic - has already been accomplished.

2. Replatform ("lift, tinker and shift") -This entails making a few cloud optimizations in order to achieve some tangible benefit without changing the core architecture of the application.

For example,

- you may be looking to reduce the amount of time you spend managing database instances by migrating to a managed relational database service such as Amazon Relational Database Service (RDS), or
- migrating your application to a fully managed platform like AWS Elastic Beanstalk.

3. Repurchase ("drop and shop") - This is a decision to move to a different product and likely means your organization is willing to change the existing licensing model you have been using.

For workloads that can easily be upgraded to newer versions, this strategy might allow a feature set upgrade and smoother implementation.

4. Refactor / Re-architect - Typically, this is driven by a strong business need to add features, scale, or performance that would otherwise be difficult to achieve in the application's existing environment. If your organization is looking to boost agility or improve business continuity by moving to a service-oriented architecture (SOA) this strategy may be worth pursuing - even though it is often the most expensive solution.

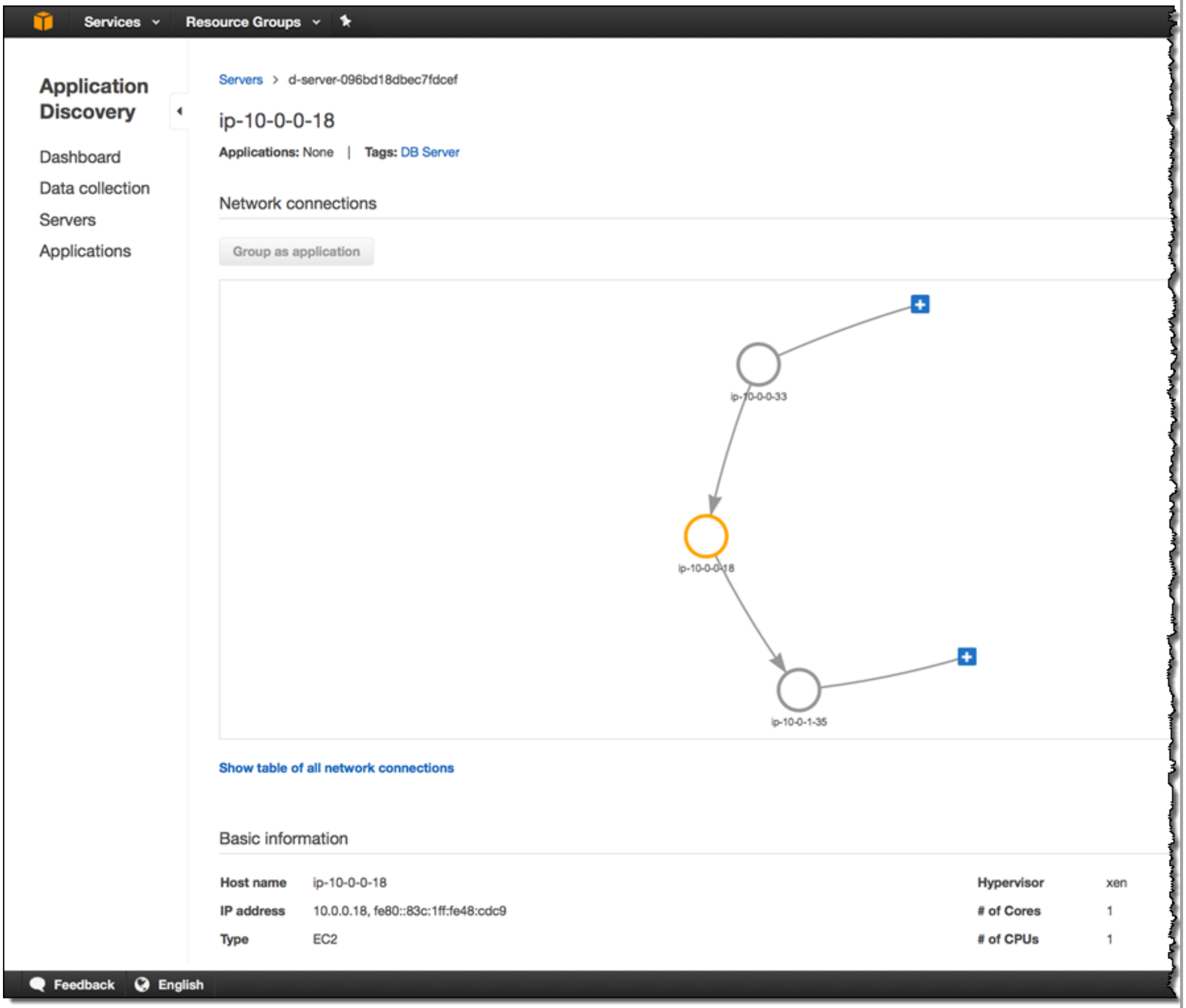
5. Retire - Identifying IT assets that are no longer useful and can be turned off will help boost your business case and direct your attention towards maintaining the resources that are widely used.

6. Retain -You may want to retain portions of your IT portfolio because there are some applications that you are not ready to migrate and feel more comfortable keeping them on-premises, or you are not ready to prioritize an application that was recently upgraded and then make changes to it again.

TOOLS : AWS Application Discovery Service

AWS Application Discovery Service helps enterprise customers plan migration projects by gathering information about their on-premises data centers.

Planning data center migrations can involve thousands of workloads that are often deeply interdependent. Server utilization data and dependency mapping are important early first steps in the migration process. AWS Application Discovery Service collects and presents configuration, usage, and behavior data from your servers to help you better understand your workloads.



Servers > d-server-096bd18dbec7fdcef

ip-10-0-0-18

Applications: None | **Tags:** DB Server

Network connections

Basic information

Host name	ip-10-0-0-18	**Hypervisor**	xen
IP address	10.0.0.18, fe80::83c:1ff:fe48:cdc9	**# of Cores**	1
Type	EC2	**# of CPUs**	1

The collected data is retained in encrypted format in an AWS Application Discovery Service data store. You can export this data as a CSV file and use it to estimate the Total Cost of Ownership (TCO) of running on AWS and to plan your migration to AWS. In addition, this data is also available in AWS Migration Hub, where you can migrate the discovered servers and track their progress as they get migrated to AWS. Hence, ***using the AWS Application Discovery Service to gather data about your on-premises data center and performing the TCO analysis*** is the correct answer.

Fault Tolerance

- Fault Tolerance is the ability of a system to remain in operation even if some of the components used to build the system fail.

- In AWS, this means that in the event of server fault or system failures, the number of running EC2 instances should not fall below the minimum number of instances required by the system for it to work properly.
- So if the application requires a minimum of 4 instances, there should be at least 4 instances running in case there is an outage in one of the Availability Zones or if there are server issues.

- One of the differences between Fault Tolerance and High Availability is that, the former refers to the minimum number of running instances.
  - Fault Tolerance = Minimum number of running instances
  - High Availability = At-least ONE instance serving the users request
- For example, you have a system that requires a minimum of 4 running instances and currently has 6 running instances deployed in two Availability Zones.
- There was a component failure in one of the Availability Zones which knocks out 3 instances. In this case, the system can still be regarded as Highly Available since there are still instances running that can accommodate the requests.
- However, it is not Fault Tolerant since the required minimum of four instances have not been met.

References:

https://media.amazonwebservices.com/AWS_Building_Fault_Tolerant_Applications.pdf

https://d1.awsstatic.com/whitepapers/aws-building-fault-tolerant-applications.pdf

Disaster Recovery - RTO, RPO

Businesses of all sizes are using AWS to enable faster disaster recovery of their critical IT systems without incurring the infrastructure expense of a second physical site. AWS supports many disaster recovery architectures, from those built for smaller workloads to enterprise solutions that enable rapid failover at scale. AWS provides a set of cloud-based disaster recovery services that enable fast recovery of your IT infrastructure and data

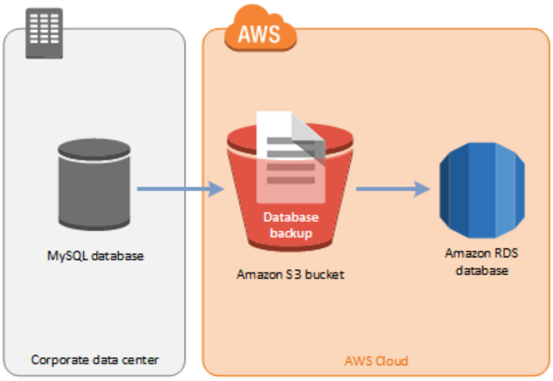
Recovery time objective (RTO) is the time it takes after a disruption to restore a business process to its service level, as defined by the **operational level agreement (OLA)**.

For example, if a disaster occurs at 12:00 PM (noon) and the RTO is eight hours, the DR process should restore the business process to the acceptable service level by 8:00 PM.

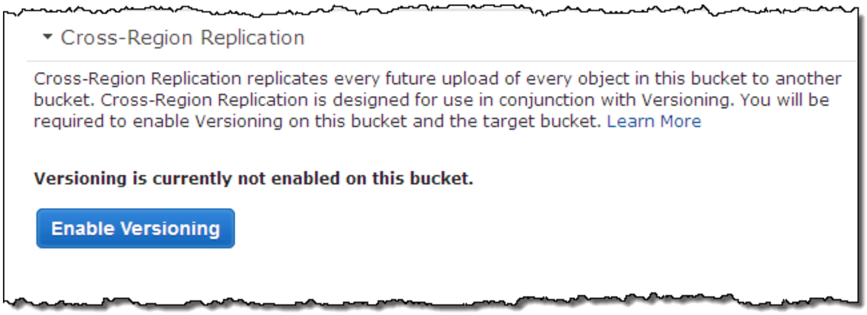
Recovery point objective (RPO) is the acceptable amount of data loss measured in time.

For example, if a disaster occurs at 12:00 PM (noon) and the RPO is one hour, the system should recover all data that was in the system before 11:00 AM. Data loss will span only one hour, between 11:00 AM and 12:00 PM (noon).

Amazon S3 is an ideal destination for backup data that might be needed quickly to perform a restore. Transferring data to and from Amazon S3 is typically done through the network, and is therefore accessible from any location.



Amazon Macie is a security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. Amazon Macie recognizes sensitive data such as personally identifiable information (PII) or intellectual property, and provides you with dashboards and alerts that give visibility into how this data is being accessed or moved. The fully managed service continuously monitors data access activity for anomalies, and generates detailed alerts when it detects risk of unauthorized access or inadvertent data leaks.



▾ Cross-Region Replication

Cross-Region Replication replicates every future upload of every object in this bucket to another bucket. Cross-Region Replication is designed for use in conjunction with Versioning. You will be required to enable Versioning on this bucket and the target bucket. Learn More

Versioning is currently not enabled on this bucket.

Cross-region replication (CRR) enables automatic, asynchronous copying of objects across buckets in different AWS Regions. Buckets configured for cross-region replication can be owned by the same AWS account or by different accounts. This is also helpful to durably store your data and for disaster recovery in the event of a region-wide outage.

References:

https://www.slideshare.net/AmazonWebServices/disaster-recovery-options-with-aws

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_PIT.html

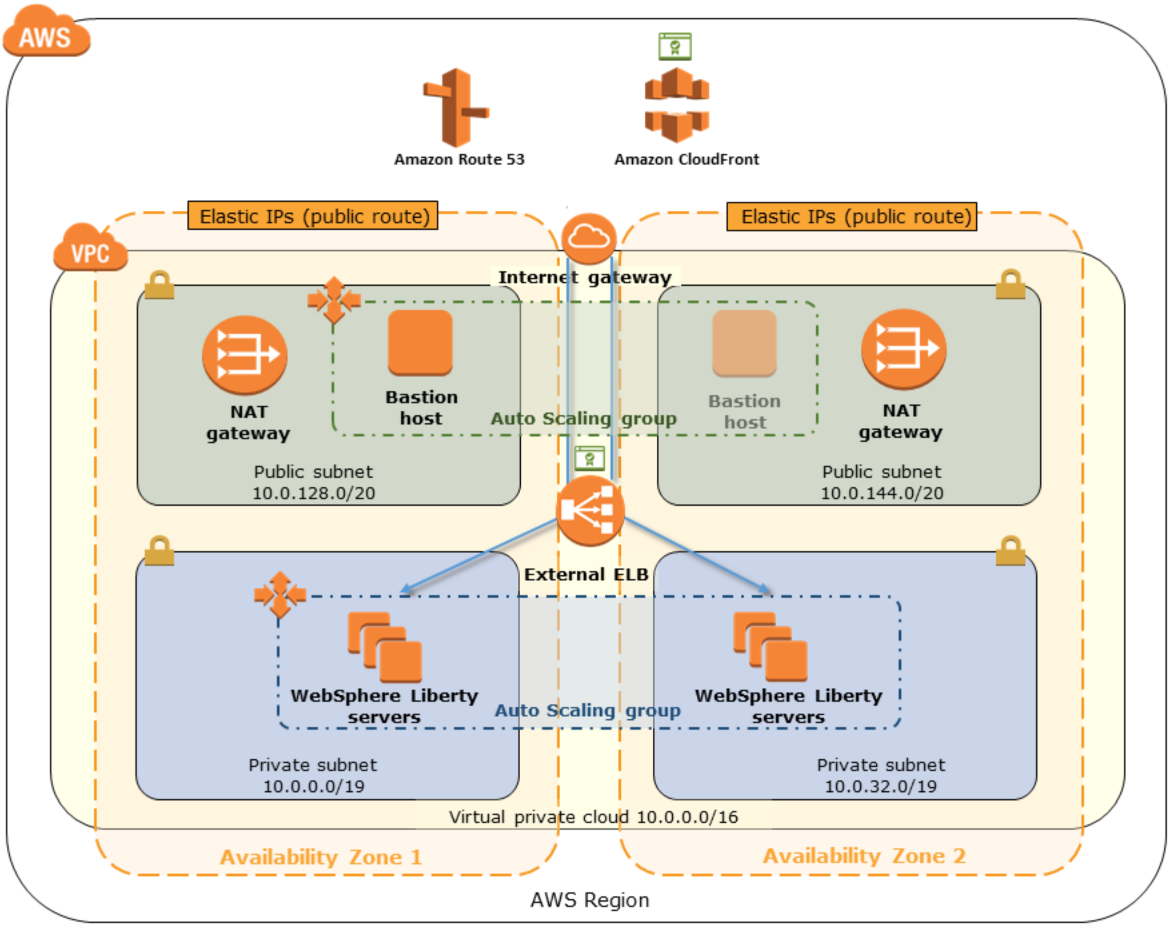
Check out this Amazon Macie Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-macie/

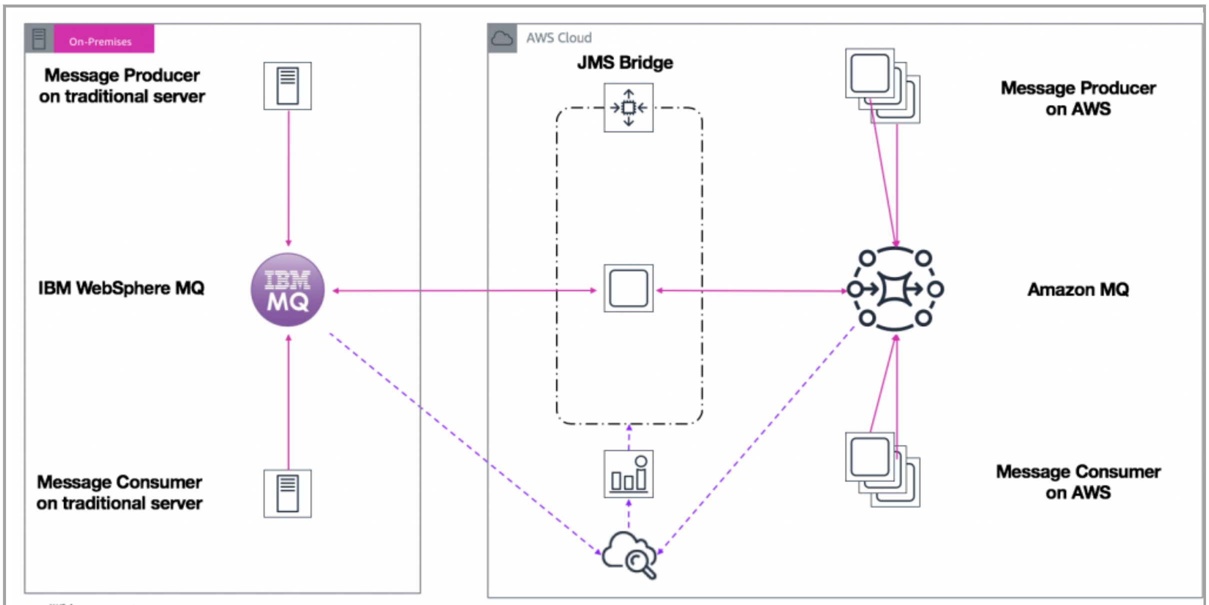
Migration - Approaches

IBM suite of products

On Amazon EC2, you can run many of the proven IBM technologies with which you're already familiar. You may be eligible to bring many of your own IBM software and licenses (BYOSL) to run on Amazon EC2 instances.


AWS
Amazon Route 53
Amazon CloudFront
Elastic IPs (public route)
Elastic IPs (public route)
VPC
Internet gateway
NAT gateway
Bastion host
Auto Scaling group
Bastion host
NAT gateway
Public subnet 10.0.128.0/20
Public subnet 10.0.144.0/20
External ELB
WebSphere Liberty servers
Auto Scaling group
WebSphere Liberty servers
Private subnet 10.0.0.0/19
Private subnet 10.0.32.0/19
Virtual private cloud 10.0.0.0/16
Availability Zone 1
Availability Zone 2
AWS Region

AWS Database Migration Service (DMS) and the **AWS Schema Conversion Tool (SCT)** can allow you to convert and migrate IBM Db2 databases on Linux, UNIX and Windows (Db2 LUW) to any DMS supported target. This can accelerate your move to the cloud by allowing you to migrate more of your legacy databases. The new Db2 LUW source adds to the existing list of relational database, NoSQL, and object store sources supported by DMS. If the database migration target is Amazon Aurora, Amazon Redshift or Amazon DynamoDB, you can use DMS free for six months.



Amazon MQ is a managed message broker service from AWS that makes it easy to set up and operate message brokers in the cloud. To migrate and re-platform your on-premises IBM MQ to Amazon MQ, you can opt for a phased approach for the migration process. You can move the producers (senders) and consumers (receivers) in phases from your on-premises to the cloud. This process uses Amazon MQ as the message broker, and decommissions IBM MQ once all producers/consumers have been successfully migrated.

Hence, the correct option is: **Use the AWS Database Migration Service (DMS) and the AWS Schema Conversion Tool (SCT) to convert, migrate and re-architect the IBM Db2 database to Amazon Aurora. Set up an Auto Scaling group of EC2 instances with an ELB in front to migrate and re-host your IBM WebSphere. Migrate and re-platform IBM MQ to Amazon MQ in phased approach.**

References:

https://aws.amazon.com/quickstart/architecture/ibm-websphere-liberty/

https://aws.amazon.com/about-aws/whats-new/2018/04/aws-dms-supports-ibm-db2-as-a-source/

https://aws.amazon.com/quickstart/architecture/ibm-mq/

https://developer.ibm.com/messaging/2018/09/26/ibm-mq-available-managed-service-aws/

https://aws.amazon.com/blogs/compute/migrating-from-ibm-mq-to-amazon-mq-using-a-phased-approach

Check out these cheat sheets on AWS Database Migration Service and AWS Server Migration Service:

https://tutorialsdojo.com/aws-cheat-sheet-aws-database-migration-service/

https://tutorialsdojo.com/aws-cheat-sheet-aws-server-migration-service-sms/

Tutorials Dojo's AWS Certified Solutions Architect Professional Exam Study Guide:

https://tutorialsdojo.com/aws-cheat-sheet-aws-certified-solutions-architect-professional/

On Premise Vmware vCenter

- Deploy the AWS Server Migration Service Connector virtual appliance on your on-premises VMware vCenter environment and use the AWS Server Migration Service for the migration process.

AWS Server Migration Service (SMS) is an agentless service which makes it easier and faster for you to migrate thousands of on-premises workloads to AWS. AWS SMS allows you to automate, schedule, and track incremental replications of live server volumes, making it easier for you to coordinate large-scale server migrations.

AWS Server Migration Service is a significant enhancement of the EC2 VM Import/Export service. The AWS Server Migration Service provides automated, live incremental server replication and AWS Console support, unlike the VM Import/Export service. Hence, ***deploying the AWS Server Migration Service Connector virtual appliance on your on-premises VMware vCenter environment and using the AWS Server Migration Service for the migration process*** is the best answer.