Database server should have cluster server, and be ready for a fail over..using RDS service this can be planned.

Loose coupling , one component should be able to function..use queue service instead of one component talking directly to other..

SWF: workflow service…can also be used …

Use self healing and rebooting for servers…if one goes down…other server comes up.

Bootstraaping and self healing..configure scripts to some startup task and apply these scripts to AMI.

## Security..

AWS has no control over OS security…running inside the instance..its the responsibility of the User..

Ram  
More ram..distribute load across machines or shared cache

More IOPs for database  
Have more read replicas for one master database..  
for multiple read replicas there is a single endpoint..the connection string is configured to point to the single endpoint..it can be a canonical name..the DNS server can be used to resolve the canonical name..  
so the developer doesnot have to worry about database server name getting changed …

The synchronization between Master and slave databases is not instantaneous..it takes 2mns…to be kept in mind ,,

Rip and Replace:  
toss bad instances if problem..

During scale up of database from a smaller instance to bigger instance (i.e more powerful server instance)..a maintanence window is required…

For highvolume data try to use in no sql…eg dynamodb  
Temprorary high volume data can be put in dynamobd and later on moved to main database or the datawarhouse service (redshift..connect jasper report)

No additional cost for hosting server in different AZs…………

## Aws import and export service

This service can be used to move data from local network to Glacier.

EC2 snapshots reside in Glacier, they are not visible though from the Glacier interface, only visible through EC2 user interface..

## Amazon Storage Gateway

Connect an onpremise software with cloud based storage..(eg S3)  
Less frequent used data can be sent to S3 and frequently used data on premise..  
a client softare or utility will be residing on the local machine..  
can be used for data backups and disaster recovery…data is to be first uploaded to S3 (data upload to S3 is free) and then jobs can be scheduled to copy data to Glacier…

**It’s only an appliance fro VMWare Infrastructure (??? Research) cannot be used on physical windows machine.**

You can write independent utilities to take backups of local users to S3 (and then to Glacier)  
Client tools are available to copy data to S3…[CloudVary]

## Amazon import/export

Use to move data across internal Amazon storages, and also on premises data to and from cloud.

### Amazon CloudFront

Caching static web pages, images , videos, ssl endpoints (also varied by user) across 40 edge locations for faster access.

Choose instances for EC2 depending on ur requirement..eg..clustering..processing etc..

## Noteremeber to write ondemand ec2 and EBS

Every

## Amazon Elastic Mapreduce

For processing huge amount of data..(eg salary processing)

## SimpleDB

Should not be used..it came earlier to Dynamo Db..(for storing huge amount of data, it’s a Rdbms but data is stored as a key value pair)..but after DynamoDb (Not rdbms) came it’s a better option for storing huge data.

## Amazon Simple Workflow Service (SWF)

Workflow service..mange orchestrization..depending on states etc…  
Coordinate processing states across distributed systems..

## Amazon CloudSearch

Used for querying indexed data…data has to be first indexed before pushing the data in..

## Amazon Simple Queue Service (SQS)

Intermediatry service for sending data from one service to other…move data between distributed component of applications..

## Amazon Simple Email Service (SES)

Bulk and transactional email sending service. The limit of sending emails..depends on how much the senders emails are blocks..better acceptance of the emails..the volume limit increases..  
good rapot with other isps..so they do not block emails coming from this service..

# Elastic Transcoder

Used for converting the media files formats…bit rates etc..

## Amazon Route 53

## AWS Elastic Beanstalk

Meant for deploying application ..it is a platform has a service..u need not have to woorry about creating EC2 instances..it manages all this internally..

Have indepent beanstalk service for managing each tier..

## Cloud Formation

Rapid deployment template..creat a cloud deployment template..can be used for standard cloud deployment architecture across different regions..

## Data Pipeline

Service for reliable transfer of data across different storage services..

## CloudHSM

Secure storage for Cryptographic keys.

## Security

One can create Groups,,associate users with groups, policiies is a set of permission…

Blogs.aws.amazon.com/security

Security Token Service (STS): this can be used to generate temprorary tokens…on the fly…instead of hardcoding password (for eg accessing the S3 service) in the config files.

Cross Account management Buckets: Trust relationship can be created between 2 different accounts, such that a user of one account can access the instances of another account if required.

## Federated Users

Eg. You can configure windows active directory users to directly login in the AWS web link (or console),,,

Using Identity Broker..it contact the token service for allocationof temprorary credentials of the fly.

Identity brokers can be created in any language (eg. C#), sample code is given..

For eg. The intranet portal can have link which invokes the identity broker custom code which uses the identity store (eg. Active directory) to generate temprorary credentials and access AWS.

<http://docs.aws.amazon.com/STS/latest/UsingSTS/STSUseCases.html>

# Aws Services for Web Applications

## Route 53

* Use Route 53..faster resolution of dns queries. It stores dns entries for all resources like s3etc.
* You can have load balancers in different Azs ..and route 53 can be used to route traffic to both load balancers in round robing maner.
* Route 53 can be used to route the traffic to in premises server or cloud depending on the load or if the on premises goes down. Or a particular region goes down..it can route to another region server.

[www.cloudping.info](http://www.cloudping.info)

## Amazon Elastic Load balancer when traffic increases load balancer adds IP address to Route 53..but autoscaling needs to be configured in coordination with Load balancer (depending on the load or no. of IPs poping up on load balancer) to increase instances..depending on load. Basically auto scaling monitors the no. of ips generated on load balancer…it picks up the ips…triggers the instances and assigns the IP to the instances…

## Amazon CloudFront

For caching static data to edge locations  
can configure what bucket data could be cache.  
to reduce pricing one can select specified edge locations…eg. Only in Europe if the application is accessed only from Europe.

One can also configure the time limit for using the cache data.  
it can be used to reduce cost of downloading data directly from S3.

Cloud front is charged depending on the transactions of bandwidth consumed..this cost is less compared to directly downloading the file from S3.

## Amazon CloudWatch

For monitoring …alarms etc. for disk space consumed on Ec2 instance..s3 buckets etc.

Not charged if monitoring done at 5mn interval.

U can use API and create own monitoring application.

## Elastic BeanStalk

Here one can configure the new instance to be created..the Autoscaling uses this configuration to trigger the instance. This makes task easier and automated..instead of direct creating instances..this can be used by developers to deploy the application…

## Cloudformation

One can create deployment templates (with runtime parameters) which can be used to duplicate deployment in n no. of regions.It can include Elastic Beanstalk configurations. The templates are in JSOn format.

<http://aws.amazon.com/cloudformation/aws-cloudformation-templates/>

for some ready made templates..which can be customized..[you will find sharepoint templates.]

U can use templates to first test the architecture on dev..push it to test and then deploy the same template on prod (changing a few runtime parameters).

## Bootstraping

Some basic configuration tasks to be done at startup (like autoexec.bat file in windows). Use the text box…”user information”

## Cfn-init

Its another technique for boot strapping..if you are are writing cloud formation template..

Auto scaling can have its own launch config…u can define what AZ’s auto scaling will span..its generally good practice to have equal no. of server in each AZ…there can be different autoscaling policies for different AZ’s

Cloud watch **watches the** Load Balancer **and triggers** autoscaling **policy.**

**In cloud watch define triggers like processor load etc..which can be used to trigger autoscaling policy.**

# EBS Voumes

Snapshots can be taken to move data from one region to another region.

To increase ebs volume size..take snapshot of original and after increasing the size one can restore the snapshot.

## DNFS

## Instance Store

deal for storing temp.. data like cache etc…

## Amazon S3

Server side encryption..is self managed (just need to select the option while uploading file)..the key used for encryption is again encrypted by S3 (this key is rotated reqularly), this happens behing the scenes..

Security Policies or permissions can be be implement at bucket & object level.

One can to webhosting in Amazon s3 (static websites)

## Static website hosting

Can keep web pages here..but can be used with Cloudfront..to keep charges low (avoid s3 download charges)..to enable url redirecting (so that any url address eg..just yahoo.com instead of [www.yahoo.com](http://www.yahoo.com)) u can multiple buckets and the one bucket with different name can point to the original bucket. (also need to configure route 53)

Logging

Notifications

Can configure automatic movement to Glacier.

**Requestor pays** the end user can pay directly..(without partner intervention).

**Multipart Uploading ..**can break file into small chunks using certain tools..

## Cloud Front

Different distribution for normal files and streaming content.

## Caching based on query string parameter

### RDS

Patch management by AWS..customer has the option to rollback to previous version before update if any problem in Prob environment.

## Dynamo Database Table partitioning handled behind the scenes by AWS to increase performance..(no need to worry about).

Just tell the primary key..throughput (no. of reads and writes)..  
gives very high performance and low latency.

Data written would be replicated across different AZ’s automatically.

Queries allowed only against primary key..also need to specify a range key…

Composite keys supported.

Cost calculated per no. reads and writes..

If multiple AZ not selected probability of loosing one day data.

Daily maintanence task..have to be handled on own..

One can specify which EC2 instance will connect to it..

For database:[parijat@amazon.com](mailto:parijat@amazon.com)

# Application Services

## Simple Queue Service:

Moving data to move across components..eg..processing images..different steps involved ..queue can be configured for each step..

Each EC2 instance hosts one worker process which caters to one particular queue..multiple ec2 instances can be autoscaled depending on the load of queue for faster processing..

## Simple Workflow Service

Decider: Takes decions..software component containg decision logic….(code can be written only in Java)

Activity Worker: software complonet to do tasks.

## Bulk Email

Good rapot with all isp providers..  
good reporting about failed..sucessful etc..

Good performance..

Quota of email sent depends on feedback of how many mails are accepted by end user…and how many marked as spam…

## CloudSearch

Sceanario to use…search product brochures provided by suppliers in a spend management applications..

First index the brochures..and provide a search option in the web application.

## <http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create_deploy_NET.quickstart.html#create_deploy_NET.quickstart.open>

## disaster recovery

if one premises fail..start on demand instances of web & database server..

in this case u only pay for the EBS volume.

Or else route traffic to cloud only in case of heavy traffic…