**AWS Whitepaper Notes**

Contents

[**AWS Security – Introductions** 1](#_Toc30025030)

[**AWS Security – Overview Process** 1](#_Toc30025031)

[**AWS Security – Best Practices** 1](#_Toc30025032)

[**AWS Security – Lambda service overview** 1](#_Toc30025033)

[**Disaster Recover – Backup & Strategy** 1](#_Toc30025034)

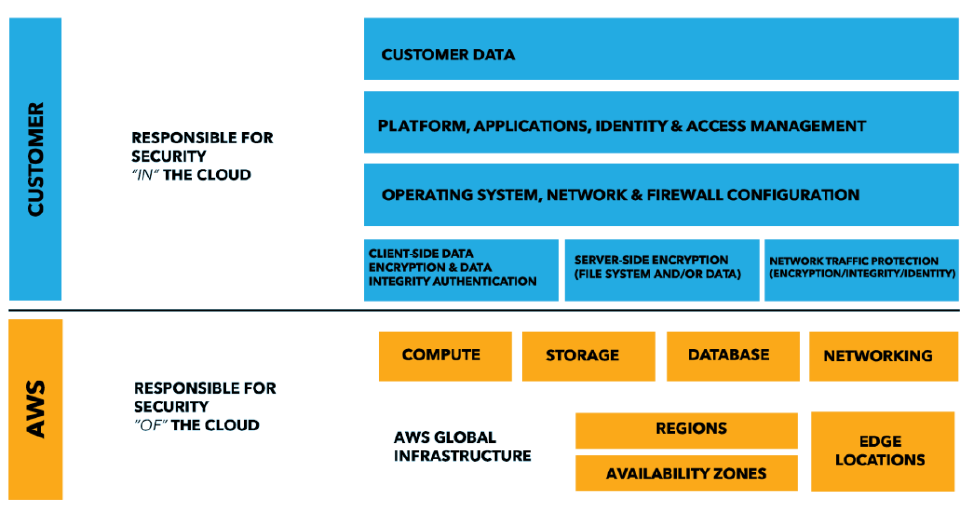
[**AWS Security Incident Reporting Guide** 1](#_Toc30025035)

# **AWS Security – Introductions**

To be added later.

# **AWS Security – Overview Process**

AWS Share Responsibility Model: Anything that put in the cloud customer is responsible for it, security of the cloud is the responsibility of the AWS. The amount of security varies between service and the sensitivity of the data. For all service basic security features are common like – SSL/TSL for data transmission, user activity monitoring etc. AWS takes care of the AWS cloud infrastructure and the security for its managed services like lambda functions, Amazon RDS, etc. For Managed services AWS takes care of the operating system patching, database patching, firewall configuration, and disaster recovery.



For customer, if they are availing IaaS service like EC2 instance then customers are responsible for

- Management of the guest operating systems (including update and security patching)

- Installation of the application utility and services

- configuration of AWS provided security firewall called as security group.

- customer is involved in the same task as they are managing their own servers irrespective of where it’s been installed.

For customer, if they are availing SaaS (AWS Managed services)

- AWS handles most of the responsibilities on behalf of the customers it takes care of the launching of the AWS manage services instances, patching of the guest operating system or databases, replicating of the database.

- Customer is responsible for managing user access and permissions for proper segregation of duties.

**AWS Security Compliance:** For AWS compliance responsibilities are also shared between customer and the AWS. AWS platform is aligned to the following security compliance standards

* SOC 1/SSAE 16/ISAE 3402 (formerly SAS 70)
* SOC 2
* SOC 3
* FISMA, DIACAP, and FedRAMP
* DOD CSM Levels 1-5
* PCI DSS Level 1
* ISO 9001 / ISO 27001
* ITAR
* FIPS 140-2
* MTCS Level 3

Along with the above list, AWS is also compliant of the following industry-specific compliance standards

* Criminal Justice Information Services (CJIS)
* Cloud Security Alliance (CSA)
* Family Educational Rights and Privacy Act (FERPA)
* Health Insurance Portability and Accountability Act (HIPAA)
* Motion Picture Association of America (MPAA)

Along with the above mentioned global and industry specific standards, AWS also provides wide range of whitepaper, reports, certifications, and other third-party attestation which can be download from the AWS site.

**Physical Security and Environmental Security:** AWS provide adequate security to protect Physical AWS datacenters – ONLY in need basis with minimum of two-factor authentications. AWS following protection for its own datacenters

1. **Fire Detection and suspensions**
2. **Power** – AWS always mating a two redundant power supplies**.**
3. **Climate and Temperature control.**
4. **Management**
5. **Storage device decommissioning:** Once a hardware (storage device) reach its end of life, AWS usages NIST (800-880) guidelines “Guidelines for Medial Sanitization” for decommissioning of the storage devices.

**Business Continuity Management : Page # 11**

# **AWS Security – Best Practices**

# **AWS Security – Lambda service overview**

# **Disaster Recover – Backup & Strategy**

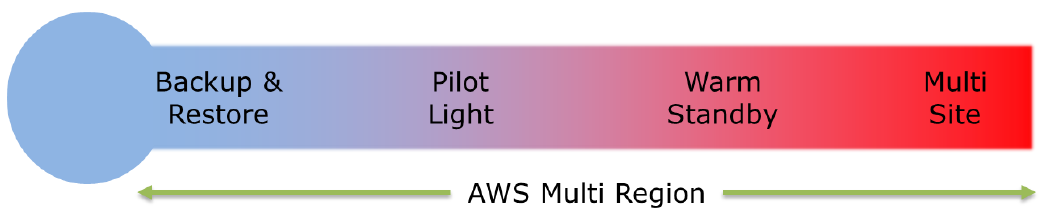
* RTO (Recovery Time Objective): The time it takes after a disruption to restore a business process to its service level, as defined by the operational level agreement (OLA). Time it takes to fully recover from a disaster (system failure)
* RPO (Recovery Point Objective): The acceptable amount of data loss measured in time. Point in time to which the data can be recovered.
* For defining designing backup strategy for backing up and restoring, identity each of the potential failure points and their business impacts, security and regulatory requirements, retention polices, RTOs and RPOs.
* Backup and Recovery strategy needs to be designed for
* File level recovery
* Volume level recovery
* Application level recovery
* Image level recovery
* Recovery strategy for different scenarios
* **Cloud Native Infrastructure** where the entire architecture is comprised of AWS services, then the ready built in feature of the AWS can be leveraged for the backup and restoration strategy. In this scenario – for file base storage leverage S3 bucket and for Volume backup one can leverage EBS snapshots stored in S3 bucket (even replicated to another region for high degree of fault tolerance). For cost optimization, incremental snapshots can be created where the first snapshot will have most of the data backed up and remains one will store the incremental changes.

Volume level Backup

* + Temporally unmounting the disk/volume to ensure consistent backup.
  + Flashing out the buffer memory in case of RAID setup
  + Using agent-based backup solution
  + Creating replica of the Primary volume (while doing this, one need to ensure that single large volume should be sufficient to address the maximum size required).

Database Backup

* + For backing up of the databases running on the EC2 instances – it can be done by creating backup of the data files using native methods /tools like EBS snapshots
  + For Large Databases which are built on RAID setup – one can offload the performance impact of primary db instance by taking backup from read replicas instead of primary db. The read replicas can have similar RAID configuration as that of the primary db OR can be set to consolidate all RAID volume into a single volume (provided required size of EBS volumes are available).
  + FOR RDS INSTANCES – manual and automated backup are available
  + AUTOMATED BACKUP FOR RDS INSTANCE: this can configure to take full daily backup at the define window set during db instance creation. Using automated backup snapshot in conjunction with the transaction logs, one can recover from any failure up-to 5 min in past. Automated backup can be retained up to 35 days.
  + MANUAL BACKUP (DB snapshot) FOR RDS INSTANCE: this can done to create a point in time backup which can be used to restore the DB or recreate a new DB instance with different endpoint.
* Image Backup
  + AMI can be created to store image which can be used to quickly provision/restore instance during any failure.
* AWS quota Elastic IP address is 5 Elastic IP addresses per account per region. Elastic IP address are regional constructs – they are confined to a single region ONLY.
* S3 durability 99.9999999999 11 9’s
* Essentials infrastructure components that needs to be consider for Backup & Recovery Strategy
* **Region:** Select multiple region, each of which are geographically separate global region, to ensure availability of one or more region during disaster event.
* **Storage:** Durable storage.
  + **S3 bucket:** Store content redundantly over multiple availability zone with high durability of 99.9999999999 11 9’s. MFA & versioning to avoid accidental delete. On premises data can be backed up into S3 through – Direct Connection, through AWS import/export portable devices, through internet. On an event of DR uploaded data can be quickly retrieved or can be used to mapped to a new instance.
  + **Glacier:** For archival storage.
  + **EBS volume:** Point in time snapshot can be created, independent of the instance lifecycle which can be stored in S3 to achieve high durability of the EC2 instance data.
  + **AWS import/export:** this feature can be used to import/export large amount of data to/from AWS using portable devices.
  + **AWS storage gateway:** This can be used to automatically backup on-premises data into AWS cloud using iSCSI protocol. One of the following variants can be used based on the need
    - **AWS Storage Gateway –** cached volumes
    - **AWS Storage Gateway –** store volumes
    - **AWS Storage Gateway –** Tape volumes
* **Compute:** Durable compute
  + **AWS EC2 instance:** compute instance can be created from store AMI (AWS machine image)
  + **AWS VM import/export:** this can be used to migrate & run on premises VM into AWS.
* **Networking:** In an event of disaster, network settings can be changed quickly to route production traffic from failed instance to DR instances. The following AWS components can help to quickly switch production traffic.
  + **Route53:** Highly available, AWS managed DNS service.
  + **Elastic IP address** – it can programmatically remap from the failed instance to new instance. Software licenses which are allocated to the MAC address , instead of remapping the elastic IP address Elastic NIC can be remapped.
  + **Elastic Load balancer** – it can automatically route the production traffic to healthy instance within the region. (*it’s a regional service, can’t route traffic outside the region*).
  + **AWS VPC:** Fully controllable data center within AWS region**.**
  + **AWS Direct Connect:**
* Database:
  + **AWS RDS –** supports, manual and automatic snapshots, multi-AZ configuration to prevent/recover from the failure.
  + **Dynamo DB –** fully managed service from AWS
  + **Redshift –** support manual and automatic snapshot which can be use to restore the failed instance within same or different AWS region.
* **Deployment Orchestration:** On an event of any failure, deployment orchestration quickly helps in restore failed instances using deployment automation and post startup installation and configuration.
  + **AWS Cloudformation:** Enable to provision AWS resources in orderly and predictable fashion. Entire configuration can be stored in a single file which can also be versioned.
  + **AWS Elastic BeanStalk:** Easy way to deploy and scale application on AWS, it’s a fully managed service on an event of any failure detection elastic beanstalk replace the underline services with healthy instance.
  + **AWS OpsWorks –** Chef base infrastructure as a code service which can be used in conjunction with AWS CloudFormation to automatically provision new stack in an event of an failure and replace the host in the newly created stack.
* Different Type of Disaster Recovery Strategy



* **Backup & Restore:** There is NO running DR site, primary site data are backed up on a remote site (*un-affected by the primary side failure*). On an event of a failure the backed-up data will be used for restoring the instances.

|  |  |
| --- | --- |
| **Preparation Phase** | **Recovery Phase** |
|  |  |

* **Pilot Light:** Minimum version of the environment is always running on the DR site. On an even of any failure, remain system can be recreated with ease and primary site can be restored.

|  |  |
| --- | --- |
| **Preparation Phase** | **Recovery Phase** |
|  |  |

* **Warm Standby:** A scale down version of the environment will be running on the DR site. In an event of any failure the DR system will be scale out to cater production traffic.

|  |  |
| --- | --- |
| **Preparation Phase -** | **Recovery Phase** |
| 1. Set up EC2 instances to mirror data 2. Create and maintain AMI for scaling up of the new instances. 3. Run AWS resources with minimal footprint. 4. Patch and update stand-by instance to keep in sync with the production environment. | 1. Scale up EC2 instance (horizontal scaling) or use a large EC2 instance (vertical scaling). 2. Manually change the DNS entry in the Route53 or use Route53 health check to route traffic to the new instance(s). 3. Scale DB layer to handle the scale up load. |

* **Active-Active:** In case of multi-site active-active configuration, fully functional production ready instances will be running in parallel on the DR site. On an event of a failure the production traffic will be route to the alternative site. **Lowest RTO, highest DR maintenance cost.**

|  |  |
| --- | --- |
| **Preparation Phase** | **Recovery Phase** |
| 1. Set the DR site equivalent to the production site. 2. Set the Route53 weight to send no/small traffic to the DR site while sending full/major traffic to the | 1. Manually change-over to the DR site 2. Use autoscaling to right size the instance fleets |

* Key factors to be consider while replicating the data
* Distance between the sites – larger the distance more latency and jitter
* Availability of the bandwidth –
* Data rate required by the application – should be lower than the available bandwidth.
* Replication technology – Synchronous Replication / Asynchronous Replication Should be done in parallel to optimize replication.
* Fallback Strategy – Once the primary site is back, the traffic needs to be routed back to the primary site from DR site. Based on the different DR strategy the step will differ

|  |  |
| --- | --- |
| **For Backup and restore Strategy** | **For Pilot light / Warm Standby / Active-Active** |
| 1. Freeze the data change on the DR site. 2. Take backup of the DR site 3. Restore Primary site using the DR site 4. Unfreeze the data change on the DR site, ensure DR site is getting backed up by the primary site data. | 1. Use reverse mirror of the data from the DR site to the primary site. 2. One Primary site is restored completely. 3. Route the traffic back to the primary site |

* Disaster readiness
  + Testing – schedule regular game-days. Ensure that game-day scenario is aligned to the real disaster scenarios.
  + Effective Monitoring and Alert
  + Keep Backup always on.
  + Evaluate licensing agreement – ensure that correct licensing option is selected.

# 

# **AWS Security Incident Reporting Guide**

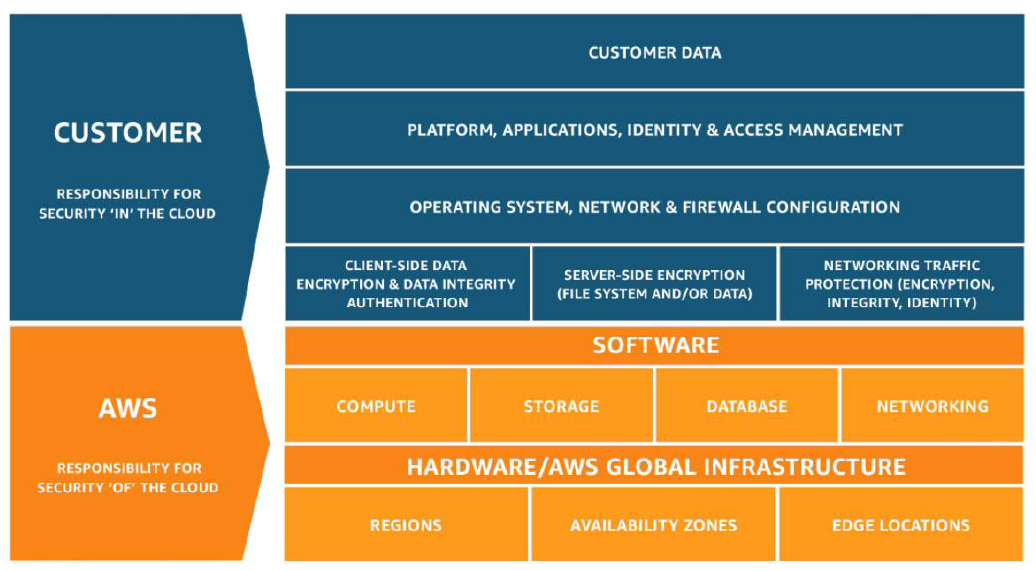
From AWS CAF (Cloud Adaptation Framework) Security prospective it consist of following four components:

* **Directive Control**: Established governance, risk and compliance model within the working environment.
* **Preventive Control**: Protect workload and mitigate threat and vulnerability.
* **Detective Control**: Provide visibility and transparency over the operations.
* **Responsive Control**: Drive remediation of potential deviation from the security baseline.

Foundation of Incident reporting are **EDUCATE**, **PREPARE**, **SIMULATE** and **ITERATE**.

**EDUCATE**

Share responsibility model.



**Design goals for cloud response**

* Establish response Objective
* Response using cloud
* Know what you have and what you need
* Use redeployment mechanism
* Automate where possible
* Chose scalable solution
* Learn and improve your process

**Cloud Security Incident domains**

There are three domains within customer responsibility where security incident can happen – ***service domain, infrastructure domain*** and ***application domain***.

**Indicator of cloud security events**

* Log & Monitoring
* Billing Activity
* Threat Intelligence
* Partner Tool – AWS Partner Network (APN) – Security Partner Solution, Security Solution in AWS Marketplace.
* AWS Outreach – AWS response to Abuse and Compromise sections.
* One-Time contact: there should be a well define / ticketing solution available which employees can use to reach out to the security when they see any abnormality related to security.

**Understanding Cloud Capability:** Understand the various services that AWS offers to detect/response to a security incident.

**Data Privacy:** Even AWS can’t see customer data.

**AWS Response to Abuse and Compromised**: AWS team proactively monitors AWS account for any suspicious and malicious activities, on such event they report and shutdowns un authorized activities running on AWS. Majority of the abuse can be categorised as

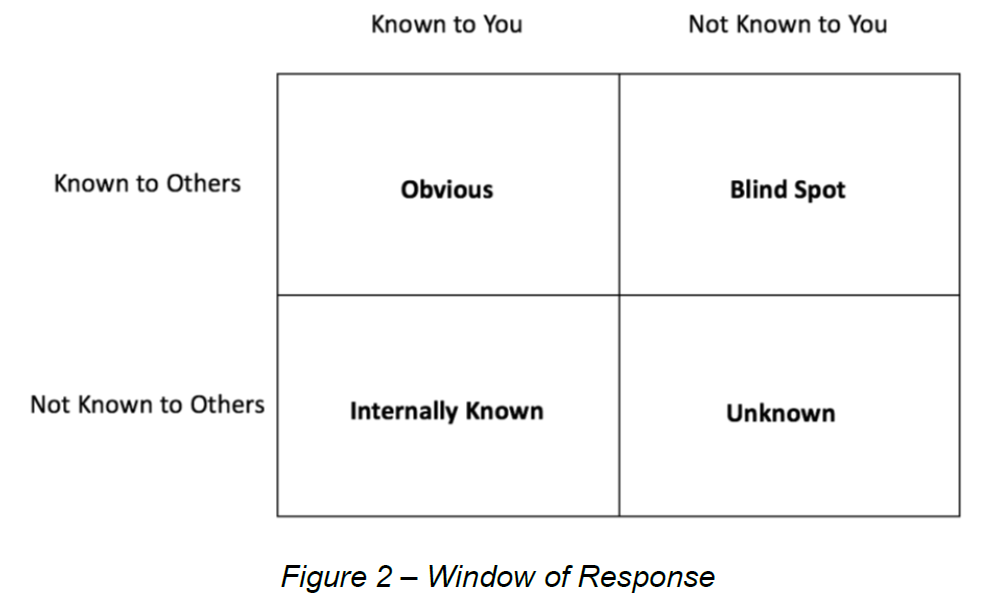
* **Compromised source**: an unpatched EC2 instance which can infect and become a botnet agent.
* **Unintentional Abuse**: Overly aggressively web crawling, this might be categorized as denial-of-services by some websites.
* **Secondary Abuse**: End users storing malicious files in hosted S3 bucket
* **False Complain**: Many a time, internet users may report a normal activity as abusive activities.

**LEARN.**

Automate the process such that, it gives humans has more time to focus and increasing security measure and spend time in corelating events, practicing simulations, device new response procedure, perform research, develop new skills, test and build new tools.

* Define Role and responsibilities
* Provide Trainings
* Define Response Mechanism
* Create a Receptive and Adaptive security culture
* Predicting response – partnered with others /share knowledge

IN 1955, Joseph Luft and Harrington Ingham created the Johari Window (window of response), a simple graph to represent the knowledge of the partner and the internal tribe. Through it was not intend to be use for security risk, the same can be easily extend for security risk.



For **Unknown** quarter, one can follow the below method to reduce the security risk.

**Defence security assertion**: Define security assertion, make it easily searchable. Start from early cloud days, then starting it late.

**Education, communication and research**: Create a cloud security expert in your team or leverage external experts, to scrutinized your environment. Create a feedback look, between the security expert and the engineering team.

**Reduce Attack Surface**: Improve defect to reduce the attack surface for unknown threats.

**Threat Intelligence**: Subscribe to continuous feed of current and relevant security threat, risk and indicators from around the world.

**Alerts**: Generate notification alert for all unusual malicious activities.

**Machine Learning**: Leverage Machine Learning to find complex abnormalities of the organization or specific persona. AWS Macie and AWS Gard Duty can be used for the same. Extend the business-centric data-lakes architecture to create security-centric data lakes, store all kinds of security information into security data lake, and leverage AWS services to derived patter from complex abnormalities.

**PREPARE.**

Prepare access to AWS Account: ensure that security response team have access to the environment where security incident was reported. Place a mechanism in place for the security teams to get access quickly. As its common practice to have multiple AWS account linked to a master account (payee account) it might be required security team to have cross account access - ensure that security team have cross account access. This can be leverage using ***service control policies***.

**In-direct access**: Security team, assist the application team/account owner to apply remediation on an event of incident.

**Direct access:** Application team/account owner deploys IAM roles for the security teams to assume such roles during incident event for applying remediation.

**Alternative Access:** Security incident responder can login into a secluded/secure account to investigate and remediate threat instead of having direct or indirect access to the actual environment.

**Automated Access:** Instead of provisioning access for the incident responder, create role specifically for the automation resources like (EC2 instance and Lambda). When incident occurs automation resources can assume such roles and act on implementing remediation. One can use AWS System Manager Run Command to run administrative tasks remotely & securely on any EC2 instance where AWS System Manager Agent is installed.

**Managed Service Access:** AWS account managed by the trusted third-party partners, to manage/implement/remediate on an event of an incident.

**Prepare Process:** Once the access is provisioned, there should be clear process defined which security incident reporting team can follow in order to investigate and remediate an incident.

**Decision Tree:** Sometimes, different actions need to be implemented based on the incident event.

**Use Alternative Account**: Security incident remediation team, may need to investigate the threat in a separate isolated account. AWS Organization can be used to create a separate forensic environment to analysed the threats. Auto infrastructure automation to create investigating environment mimicking the actual environment in the alternative account.

**View or Copy Data:** Security responders should have view access to the security logs. Appropriate IAM permission should be in place for the responders to copy point-in-time logs into investigation S3 bucket from the production bucket, in order to analyse the incident. Data can be store in S3 storage or can be archive under S3 Glacier for long term retention. One can also protect the data using S3 Glacier Vault Lock, where one can easily apply compliance base rule for long term retention.

**Share EBS Snapshot for incident investigation** – if the snapshot to encrypted make sure cross account access to the CMK (customer managed key) is provided along with the permission for copying the EBS snapshot.

**CloudWatch Logs and VPC flow logs** are store centrally. User Kinesis to process the logs from different AWS account into a single AWS account. While storing the data ensure the storage is immutable to protect the data integrity.

**Launch Resource Near the event:** Incident occurred on premises can also be investigate on the cloud environment, there are better accessibility to service to investigate & response to an incident. It may be beneficial to have a long term separate isolated AWS account for investigation, and for long term storage and legal usages.

**Isolated Resources:** There can be a need to create an isolate the resources (system) to perform forensic investigation. Best practice for launching a forensic investigation instance, Create AMIs and store the AMI or CloudFormation template so that it can be quickly provision when needed, this will also helps in standardising the forensic workstations

1. Chose relevant AMI (windows or LUNIX), for launching forensic investigation workstation.
2. Launch EC2 instance based on the AMI
3. Harden the OS, remove un wanted software packages – configure relevant auditing and Logging mechanism.
4. Install open source / private toolkit software required for investigation
5. Stop EC2 instance, create a AMI from the EC2 instance once its stopped.
6. Weekly/monthly build EC2 instance from the AMI and apply patches to the installed software/platforms.

**Cloud Provide Support**

* **AWS Support** – best practices document, whitepaper, AWS documentations, support forums etc. Chose appropriate support plans.
* **DDoS Support** (Denial of Service OR Distributed Denial of service). AWS provided AWS Shield for DoS (Standard or Advance) support. DoS Standard is free for all customers which includes standards known technique – comprehensive availability protection against well-known infrastructure attacks. User also enrol for advance DDoS protection.

**Simulate**

**Security Incident response simulation** (SIRS) helps in identifying

1. **Validate Readiness**
2. **Develop Confidence – learn from simulation and training staffs.**
3. **Follow Compliance and contractual obligations**
4. **Being agile – incremental improvement with leaser focus.**
5. **Become faster and improving tools**
6. **Refine communications and accelerations**
7. **Developing comfort with rare and unexpected scenarios**

**Simulation Steps:**

**Page#32**

Simulated examples:

**Iterate**

Create a feedback loop, to know what is working AND what is not working. One can create new procedure or update the existing procedure based on the feedback received.

Runbook

Runbook is an organization procedure, which consist of a task or series of tasks which need to be refer when there is an incident occurred. Keep re iterating the tasks to improve the core logic.

Automation

Once the core logic is defined in the runbook, one should look forward towards the automate the task(s). For automation comprehensive AWS APIs can be used

Page#36