Exam Blue Print Overview:

* AWS well architect framework.
* VPC and Direct Connect
* CI/CD pipeline
* Familiarity with AWS CLI, AWS APIs, AWS CloudFormation templates, the AWS Billing Console, and the AWS Management Console
* Whitepapers
  + Security Best Practices /Overview
  + AWS well Architect framework
  + Architecting for AWS cloud best practice
  + Continuous Integration
  + Microservices on AWS
  + Disaster Recovery
* AWS Service Documentation
  + Compute
  + Management Tool
  + Storage
  + Networking & content Management
  + Analytics
  + Database
  + Security
  + Identity and compliance
  + Application Integration (Step Functions, AWS MQ, SNS, SQS, SWF)

# Domain 1: Design for Organizational Complexity

1.1. Determine cross-account authentication and access strategy for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements).

1.2. Determine how to design networks for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements).

1.3. Determine how to design a multi-account AWS environment for complex organizations (for example, an organization with varying compliance requirements, multiple business units, and varying scalability requirements).

Key areas where one need to focus

* Identity Account Architecture: How one has to manage users across multiple AWS accounts.
* Logging Account Architecture
* Publishing Account Architecture
* Billing Structure

Identity Account (where users are created) - > sub account (environment specific to the AWS account for segregation). This can make the account management very simple. User account management remains in one account while, user can assume roles from any of the connected sub account where he is intended to work.

AWS professional

Contents

[Domain 1: Design for Organizational Complexity 1](#_Toc40031326)

[Domain 1 - Design for Organizational Complexity 2](#_Toc40031327)

[Multiple Account Strategy 2](#_Toc40031328)

[S3 ACL 2](#_Toc40031329)

# Domain 1 - Design for Organizational Complexity

## Multiple Account Strategy

Have a separate account for DEV and PRODUCTION environment, segregating the DEV & PROD environment based on region may lead to miss-management of the environment as developer may also have access to PRODUCTION environment. Have DEV and PRODUCTION created in separate account – ensure that developers doesn’t have or have minimal access to PRODCUTION environment.

When designing multiple account strategy, one needs to keep in mind the following

* **Identity Account Architecture**: Single account to manage all IAM credentials / policies.
* **Logging Account Architecture**: All logs from different account needs to be logged in a single bucket.
* **Publishing Account Architecture**: Standard gold image/services should be hosted in a centralized location (can leverage service catalogue) for the same from where all account should use the image/service to create their services in their own account.
* **Billing Architecture**: Need to have centralized billing.

Identity Account Architecture:

The identity account should NOT give any access to the users to create any kind of resources within the account. This Aws account should be used exclusively for the managing identity ONLY.

AWS License Manager – AWS License manager enforces licences across AWS accounts.

## S3 ACL

S3 bucket Canned ACL (Access Control list) – this helps to associate ACL to an object during uploading which can be used to share full access of the uploaded object to the BUCKET owner.

