At-a-Glance: How to determine the most appropriate account structure for a customer

P.AS1 As a ProServ/MRP consultant I want a well-documented flexible, initial multi-account structure so that I can quickly and easily help a customer determine the right account structure for them.

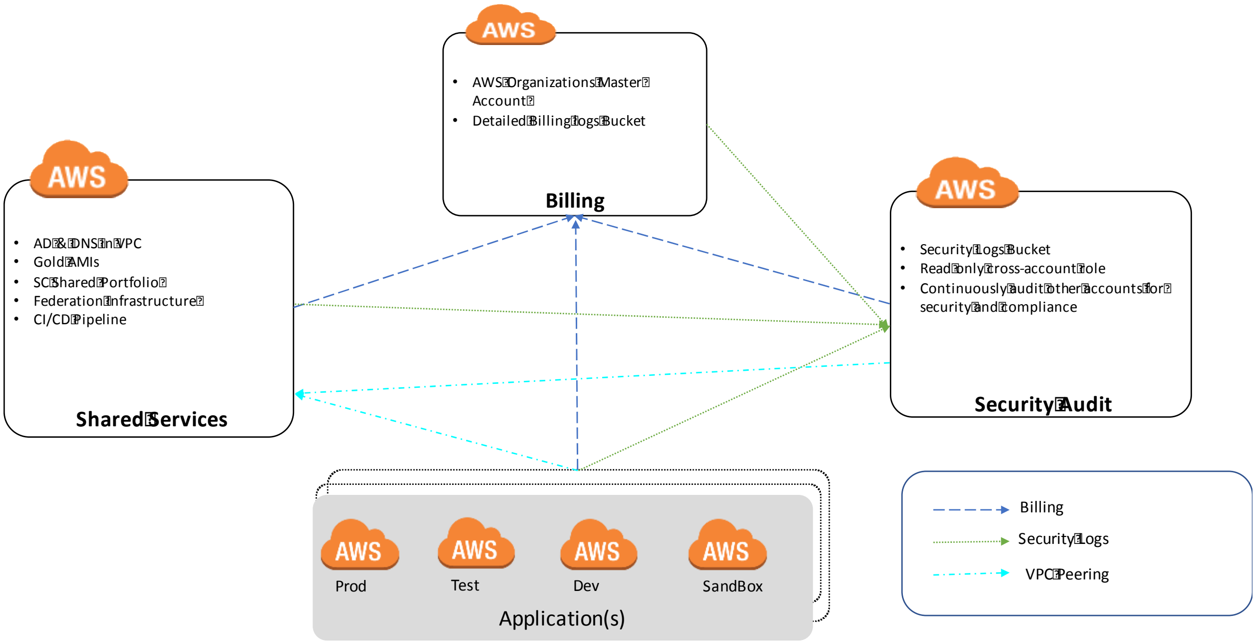
Pre-implementation Tasks

- Use a group email alias instead of an individual email address as a contact email address.
- The same email address cannot be used more than once for an account, even if the account was deleted.
- Standardize email address/naming strategy for all accounts e.g. aws+billing@company.com, aws+security@company.com, aws+app1-dev@company.com
- Pick the appropriate support plan, it can be changed later on, so start with a Basic Plan.
- Select appropriate AWS region

Customer Input

- Contact information (company name, address, phone #)
- Payment details
- Alternate contacts (name, email, phone) for Billing, Operations and Security

Account Structure



The purpose, owner and components of every account are outlined below:

- **Billing Account**: This account is the account that pays all other accounts. It can be either a single account for the entire organization or a separate account for every Line of Business (LOB) within the organization. The rest of the accounts will be linked to this billing account via consolidated billing. The billing account will store detailed billing reports for all accounts, analyze the AWS usage and spend and set billing alarms (or a budget) to provide notification in case of any threshold breach.

Additionally, the billing account serves as the AWS Organizations master account, and this master account provides the capability to create other AWS accounts. Every new account created is automatically linked to the master account for consolidated billing. The master account should host the Cross-Account Manager solution which automates account management including account creation and bootstrapping.

This account should not deploy any application workloads, it should only deploy solutions or tools for account and cost management only.

  - Owner: CIO or LOB head
  - Components:
    - AWS Organizations Master account
    - S3 bucket to store DBR logs from all accounts
    - Solutions: Cross Account Manager for account management (creating and bootstrapping new accounts), and Cost Optimization Monitor (billing and cost analysis)

- **Security Audit Account**: This account is for security and compliance-related logging and auditing activities. This account hosts the aggregated security logs from all accounts in a security logs bucket. The logs are then analyzed by Centralized Logging solution or a third-party log analysis tool. This account is also responsible for maintenance of the overall security posture of all accounts as it scans for vulnerabilities (i.e., ports open to the world) at periodic intervals. It may either take a corrective action or alert the user of any security or compliance events.

  - Owner: CISO (Security/Compliance Team)
  - Components:
    - S3 bucket to store security logs from all other accounts; Versioning and MFA delete enabled
    - Read only cross-account role to all other accounts
    - Write only cross-account role to all other accounts
    - Solutions: Centralized Logging

- **Shared-services Account**: This account hosts common services that can be shared by applications or workloads deployed in application accounts. It can be used to host golden AMIs or service catalog portfolios that are shared with application accounts. Additionally, this account can act as the global network transit center connecting different VPCs (including ones that are potentially geographically dispersed) and remote networks.

  - Owner: Infrastructure/Network Team
  - Components:
    - VPC to host AD, DNS, etc., services; peered with application accounts;
    - Service Catalog shared portfolio

        - Golden AMI Repo
        - Solutions: Transit VPC, Federation solution, CI/CD pipeline

- **Accounts by LOB, Applications, Projects**:
    - Sandbox account: This is a time and financially-boxed experimental account for developers to try new services. It does not have connectivity to on-premises or VPC peering with the shared-services account.
    - Dev/Test Accounts: These application accounts are used for development and testing. They offer network connectivity to on-premises, security baseline and VPC peering with the shared-services account.
    - Production Accounts: These accounts host the production version of applications. They have a similar setup as the Dev/Test accounts with the highest level of security controls.
    - Owner: LOB/Application team
    - Components:
        - VPC appropriate for work-load (e.g. internal-only, internal and external, etc.)
        - Customer workloads
        - Solutions to manage cost and resilience: EC2 scheduler, EBS Snapshot, Limit Monitor, Cost Optimization Right Sizing.

Implementation Guide/Steps

1. Create Billing account (e.g. aws+billing@company.com)
    Note: If using AWS Organizations, the Billing account can be made the Organizations' master account and the remaining accounts can be created from the Organizations' console in the Billing account by following these steps.
2. Signup for consolidated billing, follow these steps
    Note: This step is not required if using AWS Organizations.
3. Turn on AWS Cost and Usage reports, follow these steps, it requires:
    a. Creation of a S3 bucket e.g. Bucket name: cost-usage-report-logs-<COMPANY>-do-not-delete
    b. Grant necessary permissions on the bucket
    c. Create AWS Cost & Usage report, provide the choice to enable support for Redshift or QuickSight for cost analysis
4. Enable billing alerts, follow these steps
5. Create Security Audit account (e.g. aws+security@company.com)
6. Create S3 bucket to store CloudTrail logs from all accounts and all regions e.g. Bucket names: cloudtrail-aws-logs-<COMPANY>-do-not-delete
7. Follow these instructions to edit bucket policy to grant permission for Cloudtrail to write logs from all accounts.
    a. Change the 'myBucketName' to cloudtrail-aws-logs-<COMPANY>-do-not-delete
    b. Remove *[optional prefix] and myAccountID*
8. Create S3 bucket to store AWS Config logs from all accounts; (from all regions?) e.g. Bucket names: config-aws-logs-<COMPANY>-do-not-delete
9. Follow these instructions to edit bucket policy to grant permission for Config to write logs from all accounts.

    a. Change the ‘targetBucketName’ to config-aws-logs-<COMPANY>-do-not-delete
    b. Remove *[optional prefix] and sourceAccountID-WithoutHyphens*

10. Enable versioning on both buckets
11. Create a new S3 bucket to store access logs for the CloudTrail & Config buckets e.g. Bucket name: access-logs-<COMPANY>-do-not-delete
12. Enable logging on CloudTrail & AWS Config logs buckets with target as access logs bucket created in the prior step.
13. Setup Lifecycle policy for CloudTrail & AWS Config logs buckets to move current version to Standard - IA after 30 days, to Glacier after 60 days and expire after 425 days and previous version to Standard - IA after 30 days, to Glacier after 60 days and permanently delete after 425 days.
14. Enable MFA delete on CloudTrail & AWS Config logs buckets using S3 CLI, follow these steps
15. Create a SNS topic to stream configuration changes and notifications from AWS Config from all accounts e.g. Topic name: config-topic-do-not-delete
16. Follow these instructions to edit topic policy to grant AWS Config permission to send notifications from all accounts.
    a. Change ‘region’ to the appropriate AWS region
    b. Change ‘account-id’ to the AWS account ID of security audit account
    c. Change ‘myTopic’ to config-topic-do-not-delete
17. Create Shared-Services account (e.g. aws+shared-services@company.com)
18. Create Application accounts (sandbox, dev, test and production)
19. Follow these instructions for all accounts created above (billing, security, shared-service and application)
    a. Enable Security token-based MFA for root account. Virtual MFA applications are available for free. Follow these steps to configure virtual MFA for the root account.
    Note: This step is not required if using AWS Organizations
    b. Set complex password policy (Steps)
    c. Except for Billing account, add the account to the consolidated billing family, follow these steps
    Note: This step is not required if using AWS Organizations
    d. Enable CloudTrail in all regions, follow these steps
        a. Create a new trail e.g. trail name: security-cloudtrail-do-not-delete
        b. Select **Apply trail to all regions**
        c. Select ‘No’ for **Create a new S3 bucket**
        d. Enter S3 bucket name as cloudtrail-aws-logs-<COMPANY>-do-not-delete
    e. Enable AWS Config in all regions, follow these steps
        a. Check **Include global resources**
        b. Select **Choose a bucket from another account**
        c. Enter config-aws-logs-<COMPANY>-do-not-delete
        d. Select **Choose a topic from another account**
        e. Enter ARN for the config-topic-do-not-delete topic from security audit account
        f. Leave Prefix (optional) as blank
    f. Create cross account security read-only role, refer to these steps for general understanding of cross account roles.
        a. Create Role, e.g. Role name: SecurityAuditRole-DO-NOT-DELETE
        b. Select Role Type as **Role for Cross-Account Access**
        c. Enter AWS account ID for the security audit account for trust relationship
        d. Attach Job Function/SecurityAudit policy

20. Login to Security Audit account, follow these steps
    a. Create Policy, e.g. Policy name: SecurityReadOnlyPolicy-DO-NOT-DELETE with following JSON policy

    ```
    {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "sts:AssumeRole"
                ],
                "Resource": […]
            }
        ]
    }

    Fill the Resource section with security read-only role ARNs created in
    other accounts e.g. "arn:aws:iam::<ACCOUNT_ID>:role/
    SecurityReadOnlyRole-DO-NOT-DELETE"
    ```

    b. Create Role, e.g. Role name: SecurityReadOnlyRole-DO-NOT-DELETE
    c. Select Role Type as **AWS Service Roles -> AWS Lambda and/or AWS EC2**
    d. Attach the security read only policy
    e. Create IAM Group, e.g. Group name: SecurityReadOnlyGroup-DO-NOT-DELETE
    f. Attach the security read only policy
    g. Add security users to the IAM security group and/or deploy automated lambda/EC2 applications with security role to perform security/compliance audit checks on all accounts.
21. Setup shared service catalog in the Shared services account, detailed instructions can be found in Service Catalog Setup for MRP v1.docx
22. Setup network infrastructure in all accounts, detailed instructions can be found in Network Setup for MRP v1.docx
23. Deploy Cost Optimization Monitor solution in Billing account, follow these steps
24. Deploy following solutions in all Application accounts:
    a. Centralized Logging
    b. EC2-Scheduler
    c. EBS Snapshot Scheduler
    d. Limit monitor