Creating a basic multi-account structure

In a very basic structure you will want to focus on segregating the accounts based on environments **and workloads**. This should be a starting point for anyone transitioning from a single-account structure.

You should consider any individual group of applications, storage, systems and infrastructure components as individual workloads and group them together in their dedicated account.

- You will want to do this grouping for each environment so that you completely segregate any production workloads from the testing or development ones.

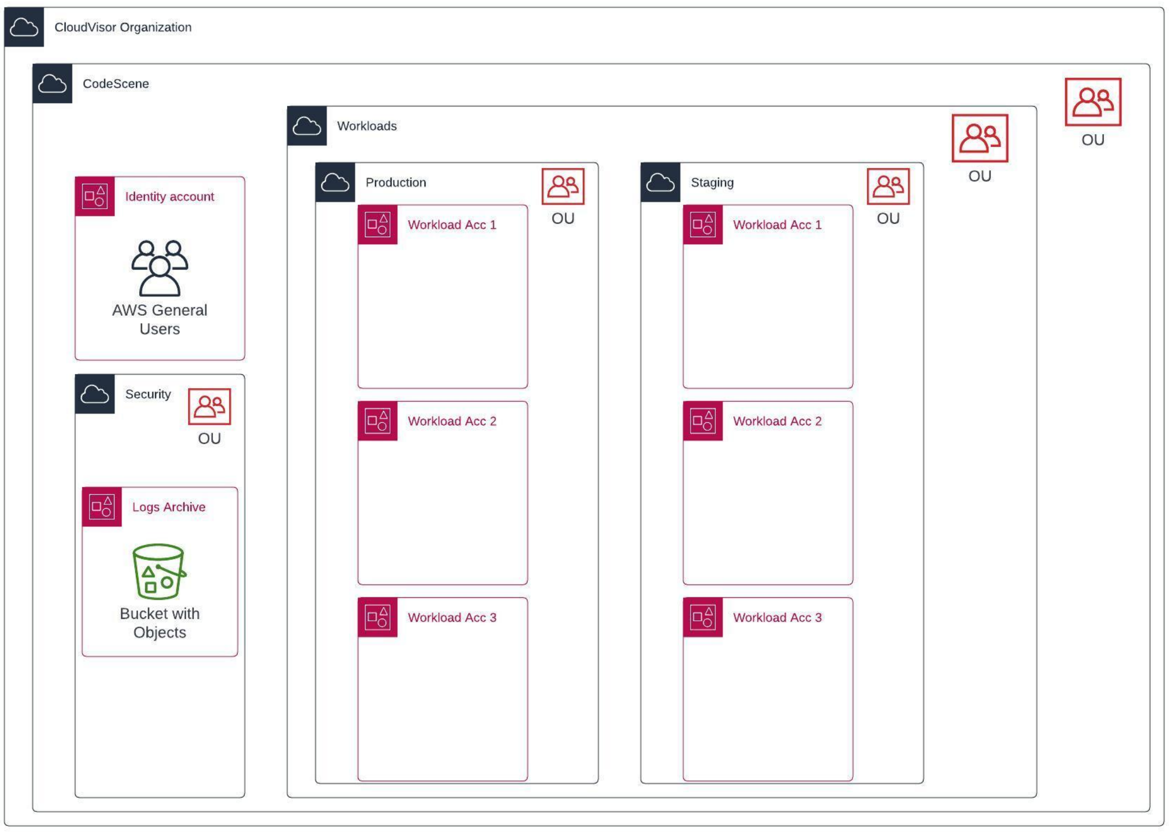
Additionally, you will want to configure a **Logs Archive** account to aggregate all the logs from all of your existing applications and infrastructure components.

- This account should store every log for compliance and historical analysis, while allowing the workloads accounts to keep only the current and relevant data required for day-to-day monitoring.
- Something to remember is that the Logs Archive account will act as a replica of your workloads account in terms of logs.
- Thus it is mandatory to make sure that proper lifecycle policies and expiration policies are in place in all accounts to avoid high logging costs.

Lastly, you will want to configure a **Management** account.

- In a regular organization, this would be your billing account and would also hold the AWS IAM Identity Center as the Organization’s IDP.
- In a reseller structure, this is going to be another member account in the organization which will hold individual IAM users which will be able to assume roles in the other accounts.
    - It will act as an **Identity** account

As a summarized structure you are looking to have something as described below:



Managing users and permissions in the Identity account:

While under a Reseller organization, the main drawback is the lack of access to IAM Identity Center. However, we can simulate a similar behavior though a dedicated IAM users account.

Implementation guidelines:

- All users will be created in the Identity account only. They will not have any access to create or modify AWS resources. They will only be allowed to assume roles in the other accounts.
  - All password requirements and access restrictions (such as MFA, expired passwords, etc.) will be configured on this account, thus securing the credentials.
  - An Administrator user will also be created to manage those users.
- All resources will be deployed in the other accounts. Roles with specific permissions will be provisioned in all the accounts.
  - All roles will have a trust relationship to the Identity account, thus allowing the users to assume them. The account level trust relationship

        is preferred in order to simplify the roles management in the workload accounts.
    - The users will be restricted in the Identity account to assume only the roles they are assigned by the Administrator.
    - Stacksets can be used to deploy the roles on all workload accounts from the Identity account, which can act as administrator for them.
- Individual groups will be created in the Identity account giving permissions to specific roles in specific accounts.
    - Users will be assigned to the groups on a per-need-only basis, thus only giving them access to the accounts and/or permissions they require.
- Each user will log in with their AWS credentials in the Identity account and then switch roles to the workloads accounts as required.
    - For programmatic access they can use their CLI credentials in the Identity account to obtain temporary credentials for a role in the workloads account.

Expanding beyond basics:

Once you feel comfortable with managing and navigating the basic setup, it would be time to expand to additional areas of segregation.

The next area of interest would be to centralize all the Security tools and configurations into their own separate account to enhance and simplify visibility.

- This should include the configuration of services such as CloudTrail, Config, GuardDuty, Security Hub, etc. The account will act as an administrator for all the other accounts and aggregate all the findings into a single interface.
- Additionally, this will provide the administrator better control for making sure that nobody changes the security settings across the organization's accounts. It will also simplify the process of auditing any issues, as all the tools are in a single place.
- The **Security Tooling** account should be placed in the **Security OU** alongside the Logs Archive account.

Optionally, you may add a new **Sandbox OU:**

- The accounts in this OU are meant for testing different services and configurations without impacting the development and release cycle of the main workloads.
- They can be used for PoCs or Demos and will most often have decreased security to allow experimentation.

Last on this section would be the **Shared Services** account:

- This account is meant to centralize all the tools and systems that are shared across your resources.
- Some example use cases include:
    - Placing the CI/CD pipelines in this account in order to avoid duplicated resources facilitating the same scope.
    - Having your monitoring and alerting solution deployed here to navigate any dashboards and alerts in a single place.
- This account should be placed in a new **Infrastructure OU**

Advanced multi-account structure:

Once everything else is in place, we should probably take a look at centralizing the Networking.

- In a nutshell, you should have all the Networking components (VPCs, Subnets, etc.) in a separate account.
    - All connections between these networks should also be managed here.
    - Optionally, any VPNs will also be configured in this account.
- All the networking resources will be shared with the workloads accounts, thus giving access to the network.
    - The application engineers can use AWS to manage and deploy their systems
    - The networking configuration is fully protected and handled exclusively by delegated administrators in the Networking account only.

Detailed reading and resources:

Organizing Your AWS Environment Using Multiple Accounts
[https://docs.aws.amazon.com/whitepapers/latest/organizing-your-aws-environment/design-principles-for-your-multi-account-strategy.html]

Delegate access across AWS accounts using IAM roles
[https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html]