TGIX

AWS Multi-Account Strategy For Highly Regulated Industries

Account Architectures that Alleviate Security Risks for Highly Regulated (and Other) Industries

An In-the-Trenches Viewpoint

Table of Contents

AWS Multi-Account Strategy For Highly Regulated Industries

AWS Multi-Account Strategy For Highly Regulated Industries

Account Architectures that Alleviate Security Risks for Highly Regulated (and Other) Industries

There's a plethora of regulatory compliance issues surrounding most businesses today. These issues directly impact decisions about the technology infrastructure used by the business, and mandate stringent controls for production environments – including enhanced security, logging, and access policies. In past years, businesses that operated in AWS implemented these controls on production workloads that were isolated in their own VPCs within a single AWS account that contained non-production VPCs as well. This intermingling of environments within a single account was a viable strategy for satisfying compliance requirements; however, managing the access controls and isolation required between the environments was challenging because the required controls didn't (and still don't) exist in AWS. The slightest of misconfigurations could expose production infrastructure and sensitive information to tens or hundreds of staff members who only needed non-production access, thereby posing a significant security risk.

Designing a multi-account strategy for your AWS infrastructure can significantly alleviate these risks. AWS provides dynamic services and enhanced networking constructs such as AWS Transit Gateway, AWS Organizations and AWS IAM Identity Center to architect, provision and manage such an integrated environment.

This white paper presents a standard reference architecture based on AWS's Well-Architected Framework, along with best practices that Tgix has employed with a number of clients to help them implement and transition to highly secure and robust multi-account environments.

Core Multi-Account Architecture

At minimum there are 4 AWS accounts in our best-practices configuration that constitute the core multi-account architecture.

Master

Master account running AWS Organizations:

- Provides consolidated billing functions
- Enforces governance through Service Control Policies (SCPs) across all accounts in the environment

Production

Production account with restricted access:

- Houses all production systems
- Provides enhanced security/ availability features

Non-Prod

Non-Prod account available to dev teams:

- Houses Dev, SIT, UAT environments
- Flexible architecture to spin up other environments on demand

Services

Infrastructure/shared services account:

- Network services: AWS Transit Gateway hub, AWS Directory Services, Route 53 (DNS), VPN
- Deployment services: Git, Jenkins, other CI/CD tooling

The **Master** account provides the central governance and billing/payment functions for all the other accounts and needs to be provisioned first. Basic account setup needs to be followed by the creation of a new organization through the **AWS Organizations** console. Consolidated billing should be subsequently enabled to ensure a single bill for all accounts and services under the organization. The Master account needs to be highly secure and stand on its own. No other computing resources or network connectivity with the other accounts should be configured.

Some examples of guardrails that can be set up through Service Control Policies (SCP) within the realm of AWS Organizations include the following: restricting privileges of admin users, tightening password policies, and enforcing MFA across all the member accounts in the organization.

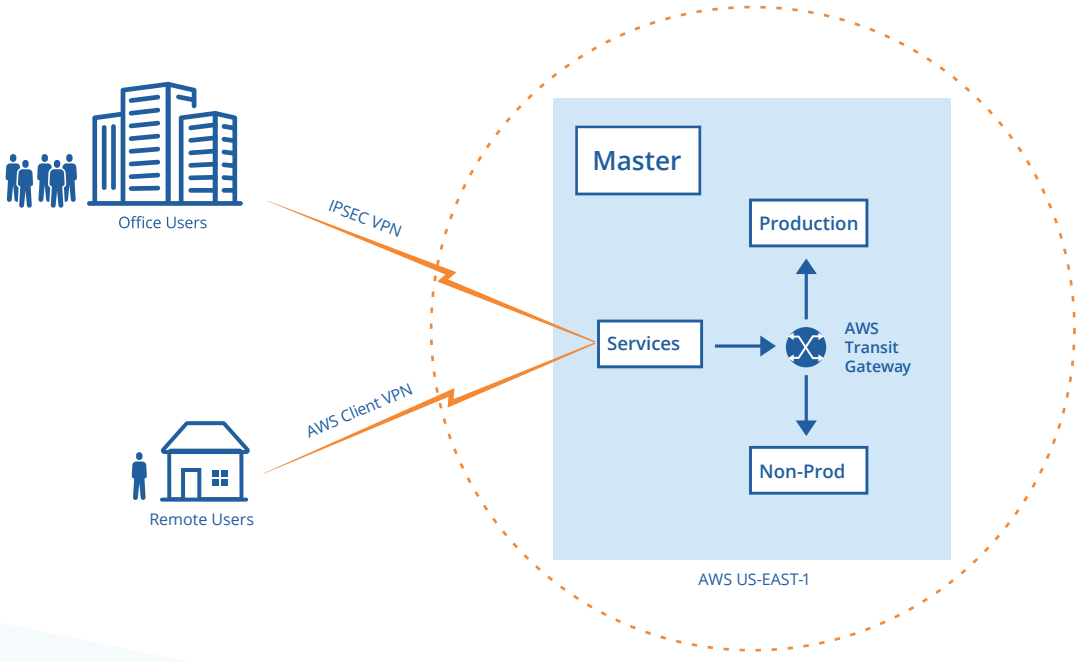
The remaining 3 accounts in our reference architecture – Production, Non-Prod, and Services – should be configured via invitations from the Master account itself, which gets them set up as member accounts in the organization. These accounts will need to communicate in myriad ways for shared access, data, services and automated pipelines. In our multi-account environment, this sharing is enabled via the network and routing components provided by the **AWS Transit Gateway.**

The **Production** account holds all the production systems and assets. Along with the Master, this is the account that auditors (and hackers) will be most interested in, thanks to potentially sensitive data (e.g., PHI, PII) that may be housed here. As such, access to this account should be limited to trusted admins, and tight security controls need to be put in place.

The **Non-Prod** account by contrast may permit access to a much larger team of people, including junior admins, devops and development staff, as this account would house various test and development environments. Care needs to be taken that no live production data finds its way into this account.

The **Services** account is the hub in the hub-and-spoke design that comprises the Transit Gateway architecture, with VPNs and routes being defined here. This architecture ensures that networking resources are consolidated and creates a "closed-loop" isolated environment with minimal external exposure and in-built internal security, e.g., controlled routing access between Production and Non-Prod accounts. Other services such as DNS and delegated IAM Identity Center (i.e. AWS SSO) services should be configured here, along with shared services such as Git and Jenkins.

High-Level Multi-Account Architecture



Businesses needing more granularity in their infrastructure can consider the following:

- Non-Prod may be further split into separate **Dev** and **Stage** accounts to reflect different environments.
- Deployment and DevOps related services may be split up into a separate **DevOps** account.
- A separate account could be set up for centralized **Log** aggregation and management.
- A separate **Sandbox** account may be created for folks to try all the radical new services that AWS seems to introduce on a daily basis; such an account may provide everyone with admin privileges and impose minimal controls outside of cost monitoring.

All these, and more, can be tied into the multi-account infrastructure.

Account Provisioning and Configuration Best Practices

AWS Control Tower can be used to provision the accounts in your multi-account infrastructure, and implement some of the guardrails described in the sections above. However, it does not integrate the core routing and federated access functions that are provided by components like the Transit Gateway. Additionally, Control Tower is much more rigid and restrictive in terms of the structure it imposes, versus the flexible mechanism described in this whitepaper that allows you to integrate both existing legacy as well as new accounts.

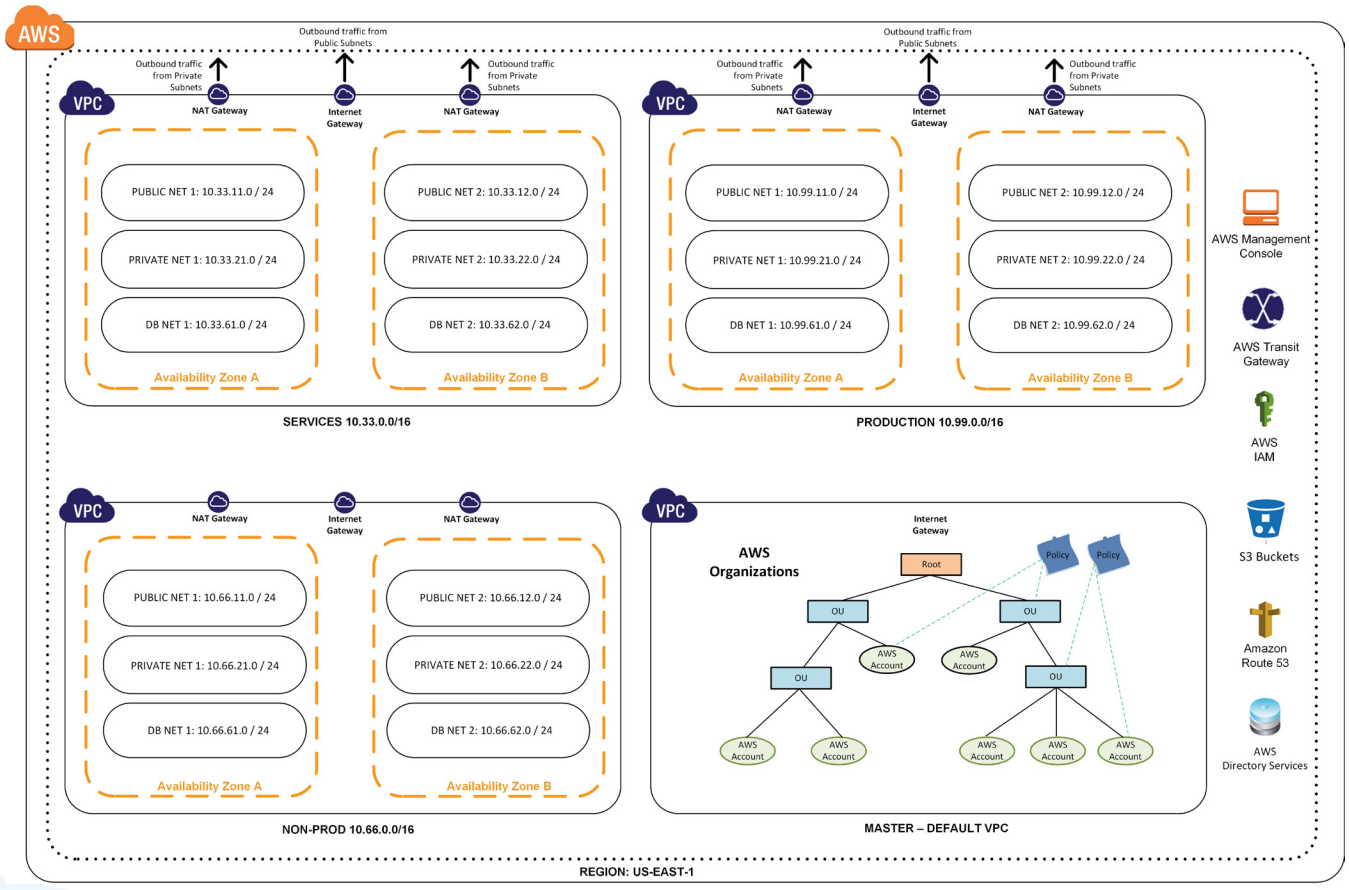
Here are some other considerations to keep in mind for configuring new accounts:

- Create a unique URL for the AWS Access Portal login
- Set up stringent password policies, and enforce MFA
- Ensure that VPC Flow Logs, CloudTrail and Config are enabled and recording
- Create consistent SNS topics & subscriptions across all accounts
- Create and ensure consistent policies and use for KMS keys
- Ensure that appropriate logging, monitoring and alerting has been set up
- Enable AWS GuardDuty for comprehensive threat detection
- Define and ensure consistent use of tags for all resources

Standardized Intra-Account Architecture

Our recommended **VPC** architecture within each account comprises of a **Multi-AZ** setup across 2 Availability Zones. This architecture ensures redundant Internet and NAT gateways. Within the VPC, we recommend a 3-tier architecture of public, private and database subnets, split into Class-C address space. The 3-tier architecture allows for more effective NACL restrictions to be configured across the subnets if this is desired. Maintaining a standardized architecture ensures a consistent application of routes, Security Groups, etc. The entire VPC-based infrastructure can be consistently defined and set up (or torn down) with the push of a button using Terraform or AWS's CloudFormation.

Standardized VPC Architecture Across Multiple AWS Accounts



Network Architecture: VPN and Routing

A site-to-site IPSEC VPN configuration with redundant tunnels may be required between the enterprise locations (HQ, other colo facilities) and the AWS environment; this VPN would terminate at the Transit Gateway in the Services account. There may additionally be a point-to-point VPN configured with one of either OpenVPN (EC2 instances in the Services account) or an AWS Client VPN implementation.

Even with a multi-account strategy, the AWS accounts may require a large number of VPCs to maintain additional isolation, e.g. between different tenants, or different business functions. It is best to be able to forecast this need, and design the architecture with CIDR blocks that a) do not conflict with other IP blocks being used within the enterprise network, and b) can be aggregated and summarized for efficient routing.

IDMS Integration: AWS IAM Identity Center, Directory Services, Okta

Managing access control at the individual account level using IAM is not a practical option. The larger the environment, the more unwieldly it would be to effectively maintain this across multiple accounts. An IDMS strategy needs to be determined at the onset. There are several choices, with the gold standard being Okta for all the different integrations that it provides and ease of management. Another option is AWS's IAM Identity Center, previously called AWS SSO, which needs to be configured in the Organizations management account (Master). This provides for centralized management of credentials and access to multiple AWS accounts and applications through a unique access portal. Management functions should be delegated to the Services account so as to not compromise the security of the Master account.

For businesses that rely on LDAP or Active Directory as their primary source of identity and access management, AWS Directory Services provides a redundant, fully-managed Microsoft AD compatible SaaS-based service. This can be configured in the Services account, and integrates with Directory Connectors that would be configured across the rest of the accounts to provide centralized access control and granular user/group management.

TGIX

In Short: 5 Steps To Consider for Your AWS Multi-Account Strategy

1. A core architecture that includes at least a master, prod, non-prod, and services accounts.
2. Follow best practices for account provisioning and configuration to ensure manageability and security.
3. Use a VPC architecture that includes separate public and private subnets set up in a multi-AZ environment.
4. Use facilities like AWS Transit Gateway to ensure a robust network architecture.
5. Implement a strategic IDMS integration to manage access control

Summary

The standard reference architectures and best practices presented in this whitepaper lay the foundation for creating a secure and robust environment, which can be easily expanded to include multiple other accounts over time. The concepts discussed here need to be combined with the general principles from our **Security Architectures Whitepaper** for a holistic solution that can be used by highly regulated industries and all others alike.

TGIX

Tgix is a certified Advanced Consulting Partner with AWS and has worked with many organizations on their regulatory compliance issues, secure cloud architectures, and support.

Contact us for a free evaluation and recommendations for a better architected and managed environment.

W: tgix.com
P: 212-918-5000
E: info@tgix.com