

PlayGround Assessment Outline - Target State Delivery

Business Summary

- Cost Management, Governance, Account Overview, Access Controls
 - Consolidate Account Types, Enforce practice for managing account services requests
 - Create 3 Accounts (Account Services, DevOps training, and Altimetrik Platform)
 - Run analysis on overall cost across 3 accounts - over the past 4 quarters to determine inefficiencies.
 - Audit and process management for DevOps Training / Account services and Platform to ensure user account lifecycle is properly managed
 - Document and roll-out the process for Accounts leveraging Account Services and provide consolidated monthly report
- Product, Service Offerings, Catalog of Use (Platforms)
 - Application products vs. AWS services vs. Network & Infrastructure components matrix
 - Consolidation of offerings utilized to streamline usage
 - Logical Diagram - business application flows for Altimetrik Platform and Account Services
- Target State / Transformational Roadmap
 - Detail Summary of 3 - 6 month plan for activities to be delivered to reduce cost, improve utilization and developer productivity
 - Determine PlayGround improvements related to process, build, package, publish, deploy, container, and cloud services

Engineering Summary

- Infrastructure Blueprint / Topology - Current State
 - Create diagram outlining the infrastructure architecture / platform
 - Breakdown across dev / qa / prod (accounts, VPCs, subnets)
 - Cross-analysis of network, system, storage, platform, AWS services for both engineering / product
- Platform / Foundational / DevOps Services
 - Adoption of automation to enable IAC
 - Adopt Terraform / CF to construct the network / infrastructure in automated fashion for RDEP/PlayGround
 - Review and structure the DevOps tools (Bitbucket, JirArtifactory, etc)
 - Analyze current VM types and # of instances for DevOps tools
 - Consolidate and validate configurations / license model / integration points
 - Foundational Services - via AWS
 - Consolidation and refactor EC2 usage and adoption model
 - Compute Instances
 - Key Pair Management
 - EBS
 - Security Group
 - Review and Remove permission to create Workspaces
 - Disable role to create / modify workspaces
 - Create rules to notify when workspace is being invoked post May 1st
 - IAM review and cleanup
 - Limit number of admin users
 - Construct standard userid format
 - No shared accounts (read-only)
 - Automated clean-up - account removal, account expiration, activity
 - Remove direct access (via Roles / Policies)
 - Certificate Manager
 - VPC
 - CloudFormation
 - CloudFormation templates are configured, but not confirmed if they are being utilized
 - S3
 - Leverage different storage type (IA) to reduce cost of items over 30 days
 - Cleanup job to remove buckets / files that don't need to be there
 - Bucket Policy enforcement
 - Tagging
 - Apply standard tagging model for all resource types
 - API Gateway
 - Cleanup of API Gateway instance that is not in used
 - NAT Gateway
 - Currently utilizing 4 NAT Gateways - need to determine if this required
 - Network ACLS
 - Validated the access control list / ports defined across the subnets
 - Virtual Private Gateway
 - S2S VPN
 - Determine the use case for utilizing VPN
 - EIP
 - Determine use case for EIP and identify if required / as we have IPs which are not being utilized
 - Route Tables

- Customer Gateway
 - Determine the use case for utilizing Customer Gateway and determine if needed
 - LightSail
 - Determine the use case for utilizing LightSail and delete if not required
 - RDB
 - Analysis of RDB databases utilized, determine if required
- Process Flows, Maturity Model, Metrics and Measurement (proactive)
 - Delivery Model (Sprint use of devOps / cloud)
 - Release Train Process (certification, environment strategy, deployment strategy)
 - Metrics - platform components, tool utilization, jira flows
- Agile / DevOps Adoption Model
 - Source Control Management
 - Repo Structure
 - Branching Model
 - Merge / Pull Request workflow
 - Artifact Versioning
 - Build Model
 - Build Definition and Architecture
 - Build Dependency / Lifecycle Management
 - CI capabilities embedded into the pipeline
 - Packaging Model
 - Publish Model
 - Artifact Lifecycle Management
 - Security Scans
 - Container Platform
 - Review of build / package / publish / deploy for container platform
 - Review of container orchestration and lifecycle management