**eHawk RePath - AWS Well-Architected Improvement plan**AWS Account ID: 028988088287  
  
Workload properties  
**Workload name**RePath - Initial Review - 2022018  
**ARN**arn:aws:wellarchitected:us-east-1:028988088287:workload/  
dd397a021c1a86346799494531b3f0f6  
**Description**First pass review  
**Review owner**kennythomason@ehawksolutions.com  
**Industry type**-  
**Industry**-  
**Environment**Production  
**AWS Regions**US East (N. Virginia), US East (Ohio), US West (Oregon), Canada (Central)  
**Non-AWS regions**-  
**Account IDs**-  
**Architectural design**-  
  
Lens overview  
**Questions answered**52/52  
**Version**AWS Well-Architected Framework, 2nd Jul 2020  
**Pillar Questions answered**Operational Excellence 11/11  
Security 10/10  
Reliability 13/13  
Performance Efciency 8/8  
Cost Optimization 10/10  
**Lens notes**-  
2/18/2022 © 2021, Amazon Web Services, Inc. or its afliates. All rights reserved. Page 5 of 72  
Improvement plan  
**Improvement item summary**High risk: 37  
Medium risk: 8  
**Pillar High risk Medium risk**

|  |  |  |
| --- | --- | --- |
| Operational Excellence | 9 | 1 |
| Security | 8 | 1 |
| Reliability | 7 | 4 |
| Performance Efciency | 4 | 1 |

Cost Optimization 9 1  
Opti9 Observations

Opti9 provided recommendations and essential notes in the section below. These are based on the well-architected review answers given during the workload review.

We strongly recommend segregating the Master account functionalities and creating the security-audit/log-archive and the shared-services AWS accounts in line with the AWS CIS Foundations Benchmark and the well-architected framework.

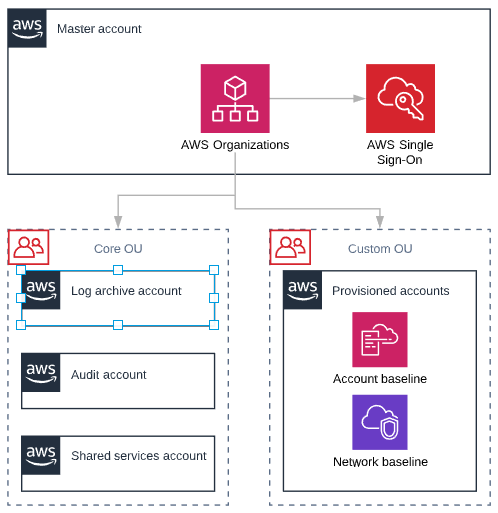
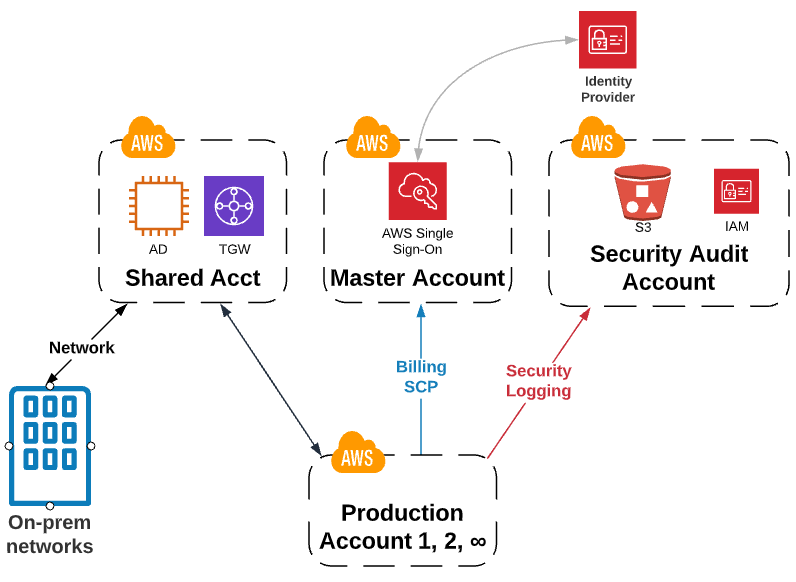
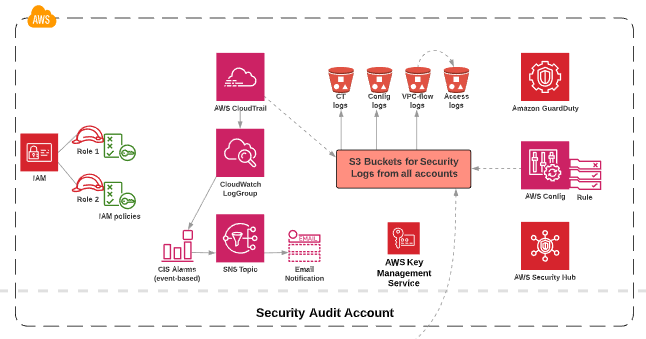
Fig 1. AWS Organization and multi-account strategy 

Fig 2. AWS multi-account strategy



Security recommendations

* The policies used should be set to the least privilege and reviewed regularly.
* Create and apply custom SCP policies
  + Isolate region(s)
  + Security controls-
    - Prevents Security CloudTrail modification
    - Prevents S3 Security bucket deletion and modification
    - Prevents GuardDuty removal
    - Prevents VPC Flow log Removal
* Configure AWS Security Hub to continuously measure the CIS AWS Foundations Benchmark and AWS Best practices.
* Use AWS Security Hub to aggregate logs into a SEIM if needed.
* Enable Amazon GuardDuty and aggregate the GuardDuty alerts across multiple accounts.
* Define controls for network protection to meet your organizational, legal, and compliance requirements.
* Evaluate and enable the logging capabilities of the services you are using, such as VPC flow logs, ELB logs, S3 bucket logs, Route53 query logs, RDS logs, etc.
* Define requirements that will help you control programmatic or automated access with appropriately defined, limited, and segregated access.
* Configure AWS SSO and use AWS Secrets Manager to manage credentials, passwords, third-party API keys if needed.
* Utilize tools such as AWS Config to monitor changes to credentials and current status. This can also be used to monitor access/secret keys age and usage and disable them or notify of rotation needed.
* Recommended AWS Config rules:
  + CloudTrail encryption enabled
  + S3 bucket public write prohibited
  + Root account MFA enabled
  + Restricted ssh
  + [IAM-policy-no-statements-with-admin-access](https://us-west-2.console.aws.amazon.com/config/home?region=us-west-2#/rules/rule-details/iam-policy-no-statements-with-admin-access)
  + [IAM-password-policy](https://us-west-2.console.aws.amazon.com/config/home?region=us-west-2#/rules/rule-details/iam-password-policy)
  + CloudTrail CloudWatch logs enabled
  + IAM root access key check
  + CloudTrail enabled
  + Multiregion CloudTrail enabled
  + Centralized guard duty
  + CloudTrail log file validation enabled
  + [cmk-backing-key-rotation-enabled](https://us-west-2.console.aws.amazon.com/config/home?region=us-west-2#/rules/rule-details/cmk-backing-key-rotation-enabled)
  + VPC FlowLogs enabled
  + Access keys rotated
  + It will inherit several security hub CIS compliance rules
* Set strict KMS key policies and store the keys in a security account with limited access. Ensure the keys are rotated regularly. Encryption should be utilized as much as possible to ensure any sensitive data is protected (including logs).
* Develop automated process that investigate different types of events, including escalation paths for incident response processes.
* Once the application is launched, run game days to prepare for incidents.

Fig 3. AWS Security Audit Account and services 

Reliability recommendations

* Monitor and manage your limits by using AWS API operations, Trusted Advisor, and AWS documentation.
* Deploy an automated limit monitoring solution, such as AWS Limit Monitor to alert you when thresholds are being approached.
* Automate healing on all layers.
* Consume all default metrics and create custom metrics for your own use such as, memory usage or disk consumption.
* Store all logs in S3, or in Amazon Glacier for longer term storage.
* Back up data. We recommend developing a DR strategy per workload and AWS service that might include:
  + Define recovery objectives for downtime and data loss. Define RPO and RTO per workload
  + Establish strategies for achieving the RTO and RPO per category. Define DR techniques
  + Perform data backup automatically
  + Perform periodic recovery of the data to verify backup integrity and processes
  + Secure and encrypt backups
* Use playbooks for unanticipated failures. Then schedule game days to regularly exercise your runbooks and playbooks
* Inject failures to test resiliency. Use open source software like Chaos Monkey or create your own failure injection code

Performance Efficiency recommendations

* Constantly improve metric collection and monitoring. As part of responding to [incidents](https://wa.aws.amazon.com/wat.concept.incident.en.html) or [events](https://wa.aws.amazon.com/wat.concept.event.en.html), evaluate which metrics were helpful in addressing the issue and which metrics could have helped that are not currently being tracked. Use this to improve the quality of metrics you collect so that you can prevent or more quickly resolve future [incidents](https://wa.aws.amazon.com/wat.concept.incident.en.html).
* Load test your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html). Deploy your latest [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) [architecture](https://wa.aws.amazon.com/wat.concept.architecture.en.html) on the cloud using different resource types and sizes. Monitor the deployment to capture [performance](https://wa.aws.amazon.com/wat.pillar.performance.en.html) metrics that identify bottlenecks or excess capacity. Use this [performance](https://wa.aws.amazon.com/wat.pillar.performance.en.html) information to design or improve your [architecture](https://wa.aws.amazon.com/wat.concept.architecture.en.html) and resource selection.
* Define compute performance requirements using benchmarking or load testing techniques.
* Collect compute-related metrics across the resources in your environment and then use a data-driven approach to optimize resources whenever is possible.
* Collect and record database performance metrics by using tools, libraries, and systems that record performance measurements related to database performance. For example, measure transactions per second, slow queries, or system latency introduced when accessing the database.
* Use the metrics your system reports to identify areas that your workload is under performing and optimize your database components.
* Evaluate AWS X-Ray to help developers analyze and debug production, distributed applications. With X-Ray, you can understand how your application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors. X-Ray provides an end-to-end view of requests as they travel through your application, and shows a map of your application’s underlying components.

Cost Optimization recommendations

* Develop policies based on your organization requirements that define how resources are managed by your organization. Policies should cover [cost](https://wa.aws.amazon.com/wat.pillar.costOptimization.en.html) aspects of resources and [workloads](https://wa.aws.amazon.com/wat.concept.workload.en.html), including creation, modification and decommission over the resource lifetime. Also develop [cost](https://wa.aws.amazon.com/wat.pillar.costOptimization.en.html) targets and goals for [workloads](https://wa.aws.amazon.com/wat.concept.workload.en.html).
* Define financial and optimization goals for accounts and workloads.
  + Gather organization requirements for the workload
  + Work with team members to identify the balance between [cost optimization](https://wa.aws.amazon.com/wat.pillar.costOptimization.en.html) and other pillars, such as [performance](https://wa.aws.amazon.com/wat.pillar.performance.en.html) and [reliability](https://wa.aws.amazon.com/wat.concept.c-reliability.en.html), for the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html).
* Implement [cost](https://wa.aws.amazon.com/wat.pillar.costOptimization.en.html) controls based on organization policies and defined groups and roles. For example, control access to regions or resource types with [IAM](https://wa.aws.amazon.com/wat.concept.iam.en.html) policies.
* Create multiple accounts to help separate [costs](https://wa.aws.amazon.com/wat.pillar.costOptimization.en.html) between different organizational units. Multiple accounts also allow more granular control of resource limits. This is already in place. We wanted to recommend it from the cost and billing perspective.
* Configure AWS Budgets in the AWS Billing and Cost Management console for each AWS account in your organization.
* Create custom reports in Cost Explorer for teams, business unites or organizational groups. Reports provide insight for current and forecast spend and usage. Reports can be customized with filters and categories; for example, across accounts, resource or usage types, and region.
* Enable AWS Cost and Usage Report to provide the most detailed view on your usage, across all accounts under the payer account, to an hourly level of granularity. The report can be loaded and processed with analytics tools such as Amazon Athena, [Amazon Redshift](https://wa.aws.amazon.com/wat.concept.redshift.en.html), or [Amazon QuickSight](https://wa.aws.amazon.com/wat.concept.quicksight.en.html), or tools from third-party providers.
* Periodically review workload resources. Use AWS Trusted Advisor to identify low and potentially unused resources.
* Perform benchmarking when you are designing a [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html). Compare the different levels of [performance](https://wa.aws.amazon.com/wat.pillar.performance.en.html) and [cost](https://wa.aws.amazon.com/wat.pillar.costOptimization.en.html) of each [component](https://wa.aws.amazon.com/wat.concept.component.en.html) against your [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) requirements, and then choose the best fit. These results are relevant as the [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) changes, as [components](https://wa.aws.amazon.com/wat.concept.component.en.html) may need to change to align with any changes or growth.
* Configure AWS Auto Scaling to dynamically adjust resourcing levels according to demand metrics, and inline with business priorities, such as [availability](https://wa.aws.amazon.com/wat.concept.availability.en.html) and [cost](https://wa.aws.amazon.com/wat.pillar.costOptimization.en.html).
* Use AWS Savings Plans.
* Configure scheduled resource management for predictable workload. For example, you can configure the AWS Instance Scheduler, which uses an [AWS Lambda](https://wa.aws.amazon.com/wat.concept.lambda.en.html) function to control the start and stop of [Amazon EC2](https://wa.aws.amazon.com/wat.concept.ec2.en.html) or [Amazon RDS](https://wa.aws.amazon.com/wat.concept.amazonrelationaldatabaseservice.en.html) instances. You can schedule additional resources of the same type, or resources with differing levels of power and throughput.
* Configure buffer-based management for workloads that are unpredictable or not time sensitive  
  A buffer queues jobs until resources become available. This enables you to provision fewer resources overall because it reduces the time they are underutilized outside of peak times. Configure [Amazon SQS](https://wa.aws.amazon.com/wat.concept.amazonsimplequeueservice.en.html) or [Amazon Kinesis](https://wa.aws.amazon.com/wat.concept.amazonkinesis.en.html) to receive and store jobs until resources become available.
* Evaluate new AWS services by subscribing to the AWS What’s New announcements.
* Review current workload architectures periodically and analyze the cost of operations.

Operational Excellence recommendations

* Identify key [performance](https://wa.aws.amazon.com/wat.pillar.performance.en.html) indicators (KPIs) based on desired business and customer outcomes. Evaluate KPIs to determine [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html) success.
* Define [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html) metrics to measure the health of [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html) and its activities. Evaluate metrics to determine if [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html) are achieving desired outcomes, and to understand the health of the [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html).
  + Publish custom metrics to CloudWatch
* Use CloudWatch Logs to search and filter log data
* Collect and analyze operations metrics by performing regular proactive reviews of metrics to identify trends and determine where appropriate responses are needed.
* Establish baselines for [workload](https://wa.aws.amazon.com/wat.concept.workload.en.html) metrics to provide expected values as the basis for comparison. Create CloudWatch Alarms.
* Alert when operations outcomes are at risk so that you can respond appropriately if required.
  + Use CloudWatch Events, alarms and invoke Lambda functions using Amazon SNS notifications.
* Alert when operations anomalies are detected.
* Validate the achievement of outcomes and the effectiveness of KPIs and metrics.
* Amazon CloudWatch dashboards can help creating a business level view of your [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html) activities to help you determine if you are satisfying needs and to identify areas that need improvement to reach business goals
* Define processes for continuous improvement.
  + Regularly evaluate and prioritize opportunities for improvement to focus efforts where they provide the greatest benefits.
  + Implement changes to improve and evaluate the outcomes to determine success. If the outcomes do not satisfy the goals, and the improvement is still a priority, iterate using alternative courses of action.
  + Your [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html) processes should include dedicated time and resources to make continuous incremental improvements possible.
* Document and share lessons learned. Have procedures to document the lessons learned from the execution of [operations](https://wa.aws.amazon.com/wat.pillar.operationalExcellence.en.html) activities and retrospective analysis so that they can be used by other teams.