Altimetrik Playground Assessment

Engagement Summary

In coordination Altimetrik Playground Team, Digital transformation team completed Cloud assessment with goal of analyzing the current cloud computing capabilities and identify improvement opportunities. We approached the engagement with a well architected cloud framework mindset – focusing on how PlayGround team may apply infrastructure provisioning practices contains security, reliability, performance efficiency, cost optimization, automation and operational excellence. The output is –

- Baseline and document finding summary of current challenges for impeding team from a frictionless cloud delivery framework.
- Recommend best practices related to Identity Access Management, Security, Operational excellence, Infrastructure as code, Environment management, Automation in provisioning/deployment, Compute life cycle management, Monitoring capabilities, Cost efficiency and Tool chain selection to develop a scalable and fault tolerant infrastructure cloud delivery model for PlayGround team to driving innovation.
- Define a comprehensive target state leveraging open source tool chain, refactored processes, and automated frameworks to accelerate cloud infrastructure delivery, instill predictability and shorten delivery time for PlayGround's cloud environments.

Our maturity assessment involves spending time with the PlayGround delivery teams to conduct a thorough review of cloud infrastructure capabilities. The aim is to jointly identify friction points, lead times and overhead and actionable steps that can be taken to:

- Provision infrastructure for technology solutions/apps faster and more efficiently – identify friction points, lead times and rework cycles
- Review of cloud infrastructure delivery models, cost efficiency, operational excellence and instill shared responsibility model for secured infrastructure practices to improve the ability of teams to scale more effectively.
- Review of Automation tool chain for infrastructure capabilities and recommend improvements related to cloud engineering practices

Key observation:

Based on working sessions across the different team leads, there were common themes identified which need to be addressed to enable a more productive, cohesive, cloud infrastructure operating environment. Improvement points have been identified in Identify Access Management, Shared responsibility model for security, Automated infrastructure provisioning, compute capabilities, monitoring practices, resource tagging strategies, environment management, measurement / metrics, and tool chain adoption.

- Cloud infrastructure team members have limited risk assessment and mitigation strategies in place while provisioning virtual machines for different technology stacks/applications. Limited governance policies for identity and access management. Network / application firewalls rules have lax controls.
- Infrastructure as a Service (IaaS) cloud delivery model used, however no automated recovery or dynamically scalable practices applied to meet demand and mitigate disruption. Confined app/network monitoring and no self-healing configuration instituted for playground solutions
- All cloud compute resources are default one for Amazon Web Services, which can be efficiently orchestrated to meet system requirements as demand changes.
- Total cost of ownership for cloud infrastructure is not clear. Automated practices to avoid or eliminate unneeded cost or suboptimal resources needed.
- Cloud infrastructure operational excellency to run and monitor cloud resources to deliver business value and to continually improve supporting processes and procedures are limited. Environments, architecture, and the configuration parameters for cloud resources not documented.

Cloud Infrastructure Delivery Framework

Current infrastructure team used Infrastructure as a service (IaaS) delivery framework for virtualized computing resources over the internet. Amazon Web Services offers a broad set of global products including compute, storage, databases, analytics, networking, mobile, developer tools, security, and enterprise applications. New services can be provisioned quickly, without the upfront capital expense. This allows PlayGround team to access the building blocks they need to respond quickly to changing business requirements.

Cloud infrastructure Security & Identity and Access Management

When cloud servers are programmable resources, cloud provider facilitate many security benefits. The ability to change your servers whenever you need to enables you to eliminate the need for guest operating system access to production environments. If an instance experiences an issue, you can automatically or manually terminate and replace it.

Starting at the network level, we can build a VPC topology that isolates parts of the infrastructure through the use of subnets, security groups, and routing controls. Using web application firewall, can help protect web applications from SQL injection and other vulnerabilities in application code.

For user access control, Identity and Access Management (IAM) enables you to securely control access to cloud services and resources for users. Using IAM, you can create and manage cloud users and groups and use permissions to allow and deny their permissions to cloud resources.

Cloud Infrastructure Operating Model

Cloud infrastructure providers provides tooling, processes, and best practices to support the transition of operational practices to maximize the benefits that can be leveraged from cloud computing.

- Applications and solutions those developed using primitive build practices, leverage the ability to manage Infrastructure as Service operating procedures which required automation for reliability.
- Limited automation of the operational processes as the supporting services, e.g. Auto Scaling and self-healing architectures.
- Solutions that are designed for cloud operations are manually developed through DevOps processes with limited resiliency considerations.

Cloud Infrastructure Cost Optimization Strategies

When we provision a cloud computing environment, optimizing for cost is a fundamental design tenant for architects. When selecting a solution, we should not only focus on the functional architecture and feature set but on the cost profile of the solutions we select.

Multiple virtual machines types used for application/solutions for PlayGround. Cloud infrastructure environment should create fewer instances of a larger instance type might result in lower total cost or better performance. We should benchmark current application environments and select the right instance type sizes depending on how PlayGround workload uses CPU, RAM, network, storage size, and I/O.

Reduce cost with continuous monitoring and tagging. Cost optimization is an iterative process. Because, each application and its usage will evolve over time, and regularly releases new options, it is important to continuously evaluate PlayGround solutions.

Cloud infrastructure provider utilities to identify cost-saving opportunities and keep our resources right-sized. We should define and implement a tagging policy for infrastructure environments. We can also use the managed rules provided by AWS Config to assess whether specific tags are applied to our resources or not.

Take advantage of the variety of purchasing options from cloud infrastructure provider - Amazon EC2 On-Demand instance pricing gives maximum flexibility with no long-term commitments. Reserved Instances Amazon EC2 Reserved Instances allow you to reserve Amazon EC2 computing capacity in exchange for a significantly discounted hourly rate compared to On-Demand instance pricing.

AWS provides fine-grained billing, which enables to track the costs associated with all aspects of PlayGround solutions.

Monitoring & Events Management

Once you have implemented your architecture you will need to monitor its performance so that you can remediate any issues before your customers are aware. Monitoring metrics should be used to raise alarms when thresholds are breached. The alarm can trigger automated action to work around any badly performing components.

- CloudWatch alarm that sends an Amazon Simple Notification Service (Amazon SNS) message when a particular metric goes beyond a specified threshold for a specified number of periods. Those Amazon SNS messages can automatically kick off the execution of a subscribed Lambda function, enqueue a notification message to an Amazon SQS queue, or perform a POST request to an HTTP or HTTPS endpoint.
- Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in AWS resources.22 Using simple rules, you can route each type of event to one or more targets, such as Lambda functions, Kinesis streams, and SNS topics.
- AWS Lambda Scheduled Events: Lambda function and configure AWS Lambda to execute it on a regular schedule.

Resiliency

Building resilience into cloud applications as a critical success factor in developing and deploying applications on the cloud infrastructure. Resiliency practices can ensure that application is available whenever users need it.

Resilience can't be an afterthought. Despite the orchestration capabilities of a Platform-as-a-Service, applications need to be designed and developed considering high availability, disaster recovery, and backup

One benefit of implementing resiliency practices is to enable customer to meet the service level agreement (SLA) established for customer enterprise application. To meet agreed SLA, PlayGround team must be sure that all applications/solutions are highly available, that you have a plan in place in case of a disaster, and that you have established backup and recovery processes for all of the critical infrastructure, services, and data needed to run your application.

Removing Single Points of Failure

Production systems typically come with defined or implicit objectives for uptime. A system is highly available when it can withstand the failure of an individual component or multiple components, such as hard disks, servers, and network links. To help us create a system with high availability, we can think about ways to automate recovery and reduce disruption at every layer of your architecture.

Redundancy can be implemented in either standby or active mode. In standby redundancy, when a resource fails, functionality is recovered on a secondary resource with the failover process. The failover typically requires some time before it completes, and during this period the resource remains unavailable. The secondary resource can either be launched automatically only when needed (to reduce cost), or it can already be running idle (to accelerate failover and minimize disruption). Standby redundancy is often used for stateful components such as relational databases. In active redundancy, requests are distributed to multiple redundant compute resources. When one of them fails, the rest can simply

absorb a larger share of the workload. Compared to standby redundancy, active redundancy can achieve better usage and affect a smaller population when there is a failure.

Finding Summary

Delivery Framework Findings

- In Playground cloud, limited testing strategy in place which can validate recovery procedures.
- No failure pathways identified for multiple applications VMs.
- Application load balancer used Scale Up/Down to manage system availability
- All Elastic Cloud Computes are On Demand types and capacity not gauged before provisioning VMs
- Manual Change applied to VMs
- Limited CloudWatch monitoring for trigger automation when a threshold is breached.
- No self-healing technique in place for VMs
- No recovery time objective (RTO) and recovery point objective (RPO), to assess a system's resiliency.
- No regularly back up for production VMs.
- Multiple large resources can increase the impact of a single failure on the overall system.

Security and Identity Assess Management Findings

- Limited NACLs rules for VPC subnets
- Current authorization is limited for each interaction with AWS resources and open logical access controls directly on resources.
- Limited firewall rules between different tier of applications
- No traceability set up – i.e. VPC Flow logs
- Manual security rules set up for each VMs
- No perimetry security rules (firewalls), between VPCs and VPN
- No Multifactor Authentication turn on for User Access Management
- No encryption for data between network or at rest endpoints
- Keypair Management System not used for encryption of access keys or secret keys.
- Amazon Simple Storage Service (Amazon S3) have open access permissions or allow access to any authenticated AWS user.

Operation and Monitoring Findings

- Multiple AWS Instances (VMs) are provisioned manually. These virtual server instances come in different families and sizes, and they offer a wide variety of capabilities, including solid state drives (SSDs) and graphics processing units (GPUs).
- Few containers virtualization in place which to run an application and its dependencies in resource -isolated processes.
- Very few functions AWS Lambda created which allow to execute code without running an instance.
- Limited regions, placement groups, and edge locations used for use case scenarios.
- Limited CloudWatch monitoring in place to raise SNS notification when performance impact
- Manual provisioning new VMs using one region which could impact the performance efficiency of cloud architecture

Cost Efficiency Findings

- Limited cost allocation tags to categorize and track AWS resource costs.
- All virtual machines are provision with On Demand type, which are most expensive in AWS cloud.
- Few services are created for POC but not deleted or stopped which increase the recurring ongoing cost for AWS cloud.
- Desktop as Service, workspaces are major contributor for the overall cost.
- Limited AWS resource tagging which give limited utilization insight for multiple VMs
- Multiple NAT Gateway Data Processing charges incurred due to route the traffic to/from S3 through the NAT Gateway
- Limited Amazon CloudWatch alarms and Amazon Simple Notification Service (SNS) notifications set up for targeted cost threshold.

Resiliency Findings

- No operations checklists to ensure if workloads are ready for production operation.
- Environments, architecture, and the configuration parameters for resources within them not documented in a way that allows components to be easily identified for tracking and troubleshooting.
- Changes to configuration also not trackable and manual
- No aggregate logs from multiple sources (e.g., application logs, AWS service-specific logs, VPC flow logs).
- No responses to unexpected operational events. This includes not just for alerting, but also mitigation, remediation, rollback, and recovery.
- Limited Trigger-Based Actions Alarms with automated actions to Start/STOP EC2 instances
- Limited monitoring in place where instances are running 24x7 which incurred cost increase month by month. Cost analysis across multiple AWS accounts suggest recent allocations for different services increased to 35-40%.

Altimetrik Digital Transformation/DevOps team in process of streamlining all cloud compute provisioning/managing/governance activities across

different teams.

Playground current state cloud platform

Security

Security is about much more than just data protection. In fact, it is a key element in a wide range of areas, some of which can be classified as follows:

Network Security: Firewalls, IDS/IPS, Web layer security, Bastion Hosts, Private & Public Subnets, External connectivity (VPN & IGW)

Data Security: Encryption mechanisms, Data resiliency & replication & Data availability and Integrity

User Access Mechanisms: API access, User Access, Federated access, and Authentication & authorization mechanisms (MFA)

Governance & Compliance: Physical data location requirements, SLAs, Contractual responsibilities & audit assessments

Monitoring & Event Management: Security assessment, Proactive threat monitoring, logging & analysis notifications

Disaster Recovery/ Business Continuity: Replication mechanism, failover techniques and minimizing service interruptions

Reliability

In the cloud, there are a number of principles that can help increase reliability:

Test recovery procedures: In the cloud, you can test how your system fails, and you can validate your recovery procedures. You can use automation to simulate different failures or to recreate scenarios that led to failures before. This exposes failure pathways that you can test and rectify before a real failure scenario, reducing the risk of components failing that have not been tested before.

Automatically recover from failure: By monitoring a system for key performance indicators (KPIs), you can trigger automation when a threshold is breached. This allows for automatic notification and tracking of failures, and for automated recovery processes that work around or repair the failure. With more sophisticated automation, it is possible to anticipate and remediate failures before they occur.

Scale horizontally to increase aggregate system availability: Replace one large resource with multiple small resources to reduce the impact of a single failure on the overall system. Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure. Stop guessing capacity: A common cause of failure in on-premises systems is resource saturation, when the demands placed on a system exceed the capacity of that system (this is often the objective of denial of service attacks). In the cloud, you can monitor demand and system utilization, and automate the addition or removal of resources to maintain the optimal level to satisfy demand without over- or under provisioning.

Manage change in automation: Changes to your infrastructure should be done using automation. The changes that need to be managed are changes to the automation.

Performance Efficiency

In the cloud, there are a number of principles that can help achieve performance efficiency

Democratize advanced technologies: Technologies that are difficult to implement can become easier to consume by pushing that knowledge and complexity into the cloud vendor's domain. Rather than having your IT team learn how to host and run a new technology, they can simply consume it as a service. For example, NoSQL databases, media transcoding, and machine learning are all technologies that require expertise that is not evenly dispersed across the technical community. In the cloud, these technologies become services that your team can consume while focusing on product development rather than resource provisioning and management.

Go global in minutes: Easily deploy your system in multiple regions around the world with just a few clicks. This allows you to provide lower latency and a better experience for your customers at minimal cost. Use serverless architectures: In the cloud, server-less architectures remove the need for you to run and maintain servers to carry out traditional compute activities. For example, storage services can act as static websites, removing the need for web servers; and event services can host your code for you. This not only removes the operational burden of managing these servers, but also can lower transactional costs because these managed services operate at cloud scale.

Experiment more often: With virtual and automatable resources, you can quickly carry out comparative testing using different types of instances, storage, or configurations.

Cost Optimization

In the cloud you can follow a number of principles that help you achieve cost optimization:

Reserved Instances: First step is to reserve instance on appropriate batches with shorter payment frequencies. Optimized reservation and payment frequency based on requirements and usage

Amazon S3 Optimization: Amazon S2 usage can be optimized by implementing a delete policy, defining object life cycle with infrequent access storage (IAS) and Amazon glacier (archival services)

Adopt a consumption model: Pay only for the computing resources that you consume and increase or decrease usage depending on business requirements, not by using elaborate forecasting. For example, development and test environments are typically only used for eight hours a day during the work week. You can stop these resources when they are not in use for a potential cost savings of 75 percent (40 hours versus 168 hours).

Benefit from economies of scale: By using cloud computing, you may achieve a lower variable cost than you could on your own because AWS can achieve higher economies of scale. Hundreds of thousands of customers are aggregated in the AWS Cloud, which translates into lower pay-as-you-go prices.

Stop spending money on data center operations: AWS does the heavy lifting of racking, stacking, and powering servers, so you can focus on your customers and business projects rather than on IT infrastructure.

Analyze and attribute expenditure: The cloud makes it easier to accurately identify the usage and cost of systems, which then allows transparent attribution of IT costs to individual business owners. This helps measure return on investment (ROI) and gives system owners an opportunity to optimize their resources and reduce costs.

Use managed services to reduce cost of ownership: In the cloud, managed services remove the operational burden of maintaining servers for tasks like sending email or managing databases. And because managed services operate

Operational Excellence

In the cloud you can follow a number of principles that help you achieve operational excellence:

Perform operations with code: When there are common repetitive processes or procedures, use automation. For example, consider automating configuration management, changes, and responses to events.

Align operations processes to business objectives: Collect metrics that indicate operational excellence in meeting business objectives. The goal should be to reduce the signal to noise ratio in metrics, so operational monitoring and responses are targeted to support business-critical needs. Collecting metrics that are unnecessary will prevent effective responses to unexpected operational events by complicating monitoring and response.

Make regular, small, incremental changes: Workloads should be designed to allow components to be updated regularly. Changes should be done in small increments, not large batches, and should be able to be rolled back without affecting operations. Put operations procedures in place to allow for the implementation of those changes without downtime for maintenance or the replacement of dependent service components.

Test for responses to unexpected events: Workloads should be tested for component failures and other unexpected operational events. It is important to test and understand procedures for responding to operational events, so that they are followed when operational events occur. Set up game days so you can test responses to simulated operational events and failure injections.

Playground transformational cloud platform actions

Security

1. IAM Roles/Policy management
2. Cross Account access management
3. AWS KMS for encryption key pairs for CLI and Application access
4. IAM role-based access for S3 Buckets
5. Encryption for Network data at transit and at rest
6. Automated security group set up for VM provisioning
7. Turn on VPC Flow logs
8. AWS Config for resource tracking activities
9. CloudWatch alert notification

Reliability

1. CloudWatch Monitoring
2. CloudFormation templates for different technology stacks
3. Enable VPC Flow logs
4. CloudTrail to monitor AWS API calls
5. Set up monitoring for recovery time objective (RTO) and recovery point objective (RPO), to assess a system's resiliency
6. Set up standard procedure to back up data, and validate backup files, to ensure it can recover from both logical and physical errors.
7. Automation in place for system to detect failure and automatically heal itself.
8. Automation in place for large resource with multiple small resources to reduce the impact of a single failure on the overall system. Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure

Performance Efficiency

1. Go global in minutes and deploy resources in multiple locations across the globe to be closer to your end users.
2. Dynamically add read-only replicas to information stores such as database systems to reduce the load on the primary database.
3. AWS also offers caching solutions such as Amazon ElastiCache, which provides an in-memory data store or cache, and Amazon CloudFront, which caches copies of your static content closer to end-users.
4. Automated triggers to avoid human error and can reduce the time to fix problems
5. Set up CloudWatch to monitor and send notification alarms, and can use automation to work around performance issues by triggering actions through Amazon Kinesis, Amazon Simple Queue Service (SQS), and AWS Lambda
6. Apply automation scripts while provisioning new VMs using new regions, edge location, services which could positively improve the performance efficiency of cloud architecture.

Cost Optimization

1. Review existing architectural decisions to ensure they continue to be the most cost effective. As requirements change, be aggressive in decommissioning resources and entire services, or systems that you no longer require
2. Managed services from AWS can often significantly optimize a solution, so it is good to be aware of new managed services as they become available.
3. Combining tagged resources with entity lifecycle tracking makes it possible to identify orphaned resources or projects that are no longer generating value to the business and should be decommissioned.
4. Set up billing alerts to notify when account cost reach to predicted overspending
5. Auto Scaling and demand, buffer, and time-based approaches allow to add and remove resources as needed. If you can anticipate changes in demand, you can save more money and ensure your resources match your system needs
6. Run trust advisory reports to identified unused resource allocation.

Operational Excellence

1. Create an operational checklist
2. Have a proactive plan for events (e.g., marketing campaigns, flash sales) that prepares for both opportunities and risks that could have a material impact on business
3. Create a security checklist
4. Resource Tracking Plan for ways to identify AWS resources and their function within the workload (e.g., use metadata, tagging).
5. Documentation Document your architecture (e.g., infrastructure as code, CMDB, diagrams, release notes).
6. Captured operational learnings over time (e.g., wiki, knowledge base, tickets)
7. Establish an immutable infrastructure so that can be redeploy
8. Automate your change procedures
9. Be prepared to revert changes that introduce operational issues (e.g., roll back, feature toggles).
10. Use risk mitigation strategies such as Blue/Green, Canary, and A/B testing.
11. Use Amazon CloudWatch, third-party, or custom monitoring tools to monitor performance.
12. Have a playbook that team can follow (e.g., on call process, workflow chain, escalation process) and update regularly.

Delivery Framework Target State

Selection:

- The key AWS service for performance efficiency is Amazon CloudWatch.
- **Compute:** Auto Scaling is key to ensuring that you have enough instances to meet demand and maintain responsiveness.
- **Storage:** Amazon EBS provides a wide range of storage options. Amazon S3 provides serverless content delivery and Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances.
- **Database:** Amazon RDS provides a wide range of database features such as provisioned IOPS and read replicas, that allow you to optimize for your
- **Network:** Amazon Route 53 provides latency-based routing. Amazon VPC endpoints and Direct Connect can reduce network distance

Review:

- AWS website are resources for learning about newly launched features and services.

Monitoring:

- Amazon CloudWatch provides metrics, alarms, and notifications.

Tradeoff:

- Amazon ElastiCache, Amazon CloudFront, and AWS Snowball are services that allows to improve performance.

Security and Identity Assess Management Target State

Identity and Access Management

1. IAM governance policies implementation to securely control access to AWS services and resources
2. IAM Roles and policies segregation by core services
3. Cross-Account IAM access management
4. Standard Lambda functional IAM roles for monitoring/deployment activities

Detective Controls

1. AWS CloudTrail set up for AWS API calls
2. AWS Config set up for detailed inventory of AWS resources and configuration
3. Amazon CloudWatch monitoring service for AWS resource

Infrastructure Protection

1. Apply perimetry security for VPC and apply standard NACL rules for each subnet
2. Set up security groups for additional security between different tiers of application VMs.

Data Protection

1. Include encryption capabilities to protect data in transit and at rest.
2. Use AWS Key Management Service to create and control keys used for encryption.

Incident Response

1. Use Amazon CloudFormation to create a trusted environment for conducting investigations for incident

Operation and Monitoring Target State

- Multiple AWS Instances (VMs) are provisioned manually. These virtual server instances come in different families and sizes, and they offer a wide variety of capabilities, including solid state drives (SSDs) and graphics processing units (GPUs).
- Few containers virtualization in place which to run an application and its dependencies in resource -isolated processes.
- Very few functions AWS Lambda created which allow to execute code without running an instance.
- Limited regions, placement groups, and edge locations used for use case scenarios.
- Limited CloudWatch monitoring in place to raise SNS notification when performance impact
- Manual provisioning new VMs using one region which could impact the performance efficiency of cloud architecture

Cost Efficiency Target State

Cost-effective resources

- The key AWS feature that supports cost optimization is cost allocation tags, which help you to understand the costs of a system.
- Use Reserved Instances and prepaid capacity to reduce your cost. AWS Trusted Advisor can be used to inspect your AWS environment and find opportunities to save money.

Matching supply and demand

- Auto Scaling allow to add or remove resources to match demand without overspending.
- Expenditure awareness
- Amazon CloudWatch alarms and Amazon Simple Notification Service (SNS) notifications will warn if you go over, or are forecasted to go over, your budgeted amount.

Optimizing over time:

- The AWS Blog and the What's New section on the AWS website are resources for learning about newly launched features and services.
- AWS Trusted Advisor inspects your AWS environment and finds opportunities to save money by eliminating unused or idle resources or committing to Reserved Instance capacity.

Resiliency Target State

Foundations

- The Amazon CloudWatch service is key to ensuring reliability which monitors run-time metrics.
- AWS Identity and Access Management (IAM) enables to securely control access to AWS services and resources.
- The system should be designed to detect failure and automatically heal itself.
- CloudFormation templates for Amazon VPC provision a private, isolated section of the AWS Cloud where you can launch AWS resources in a virtual network.

Change management

- AWS Config provides a detailed inventory of AWS resources and configuration, and continuously records configuration changes.
- Regularly back up data, and test backup files, to ensure it can be recovered from both logical and physical errors

Failure management:

- AWS CloudFormation provides templates for the creation of AWS resources and provisions them in an orderly and predictable fashion.
- Actively track KPIs, such as the recovery time objective (RTO) and recovery point objective (RPO) to assess a system's resiliency (especially under failure-testing scenarios).

Operational Excellence

Preparation

- Effective preparation is required to drive operational excellence
- Create an operational checklist that evaluate if system is ready to operate the workload.
- Create a security checklist that evaluate if system is ready to securely operate the workload

Operations

- Monitoring Use Amazon CloudWatch, third-party, or custom monitoring tools to monitor performance.
- Aggregate Logs Aggregate logs from multiple sources (e.g., application logs, AWS service-specific logs, VPC flow logs, CloudTrail).
- Alarm-Based Notifications Receive an automatic alert from your monitoring systems if metrics are out of safe bounds.
- Trigger-Based Actions Alarms cause automated actions to remediate or escalate issues.

Responses

- For unplanned operational events should follow a pre-defined playbook that includes stakeholders and the escalation process and procedures.
- Appropriately Document and Provision
- Put necessary stakeholders and systems in place for receiving alerts when escalations occur.
- Functional Escalation with Queue-based Approach Escalate between appropriate functional team queues based on priority, impact, and intake mechanisms.
- Hierarchical Priority Escalation is Automated When demand or time thresholds are passed, priority automatically escalates.

Infrastructure as Code

AWS assets are programmable, we can apply techniques, practices, and tools from software development to make our whole infrastructure reusable, maintainable, extensible, and testable.

AWS CloudFormation templates give us an easy way to create and manage a collection of related AWS resources, and provision and update them in an orderly and predictable fashion. We can describe the AWS resources and any associated dependencies or runtime parameters required to run our application. Our CloudFormation templates can live with our application in our version control repository, which allows us to reuse architectures and reliably clone production environments for application provisioning.

Automation

In a traditional IT infrastructure, we often have to manually react to a variety of events. When deploying on AWS, there is an opportunity for automation, so that we improve both our system's stability and the efficiency of Altimetrik organization. Consider introducing one or more of these types of automation into our application architecture to ensure more resiliency, scalability, and performance.

- Serverless Management and Deployment - When we adopt serverless patterns, the operational focus is on the automation of the deployment pipeline. AWS manages the underlying services, scale, and availability. AWS CodePipeline, AWS CodeBuild, and AWS CodeDeploy support the automation of the deployment of these processes.
- Infrastructure Management and Deployment

  AWS Elastic Beanstalk - This service to deploy and scale web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. Developers can simply upload their application code, and the service automatically handles all the details, such as resource provisioning, load balancing, auto scaling, and monitoring.

  Amazon EC2 auto recovery - An Amazon CloudWatch alarm that monitors an EC2 instance and automatically recovers it if it becomes impaired. A recovered instance is identical to the original instance, including the instance ID, private IP addresses, Elastic IP addresses, and all instance metadata. However, this feature is only available for applicable instance configurations. In addition, during instance recovery, the instance is migrated through an instance reboot, and any data that is in-memory is lost.

  AWS Systems Manager - Automatically collect software inventory, apply OS patches, create a system image to configure Windows and Linux operating systems, and execute arbitrary commands. Provisioning these services simplifies the operating model and ensures the optimum environment configuration.

  Auto Scaling - Maintain application availability and scale Amazon EC2, Amazon DynamoDB, Amazon ECS, Amazon Elastic Container Service for Kubernetes (Amazon EKS) capacity up or down automatically according to the conditions specified. Apply Auto Scaling to help make sure that we are running the desired number of healthy EC2 instances across multiple Availability Zones. Auto Scaling can also automatically increase the number of EC2 instances during demand spikes to maintain performance and decrease capacity during less busy periods to optimize costs.

AWS Managed Services

AWS managed services provide building blocks that developers can consume to power their applications. These managed services include databases, machine learning, analytics, queuing, search, email, notifications, and more. For example, with Amazon SQS you can offload the administrative burden of operating and scaling a highly available messaging cluster, while paying a low price for only what you use. Amazon SQS is also inherently scalable and reliable. The same applies to Amazon S3, which enables you to store as much data as you want and access it when you need it, without having to think about capacity, hard disk configurations, replication, and other related issues.

Other examples of managed services that power your applications include:

- Amazon CloudFront for content delivery
- ELB for load balancing
- Amazon DynamoDB for NoSQL databases
- Amazon CloudSearch for search workloads
- Amazon Elastic Transcoder for video encoding
- Amazon Simple Email Service (Amazon SES) for sending and receiving emails

Conclusion

When we design your architecture in cloud architecture, it is important to consider the important principles and design patterns available in current available cloud providers, including how to select the right database for your application, and how to architect applications that can scale horizontally and with high availability. Because each implementation is unique, we must evaluate how to apply this guidance to our implementation. The topic of cloud computing architectures is broad and continuously evolving.

Current AWS Accounts

- Amazon Practices : https://536285340728.signin.aws.amazon.com/console
- Amazon Platform : https://altimetrik-platform.signin.aws.amazon.com/console
- Amazon Playground : https://playground-new.signin.aws.amazon.com/console