Operational Excellence Pillar

AWS Well-Architected Framework

November 2017

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

The focus of this paper is the operational excellence pillar of the Amazon Web Services (AWS) Well-Architected Framework. It provides guidance to help you apply best practices in the design, delivery, and maintenance of AWS environments.

Introduction

The AWS Well-Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS.[1] By using the Framework you will learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. It provides a way for you to consistently measure your architectures against best practices and identify areas for improvement. We believe that having well-architected systems greatly increases the likelihood of business success.

The framework is based on five pillars:

- Operational Excellence
- Security
- Reliability
- Performance Efficiency
- Cost Optimization

This paper focuses on the operational excellence pillar and how to apply it as the foundation of your well-architected solutions. Operational excellence is challenging to achieve in traditional on-premises environments where operations is perceived as a function isolated and distinct from the lines of business and development teams that it supports. By adopting the practices in this paper you can build architectures that provide insight to their status, are enabled for effective and efficient operation and event response, and can continue to improve and support your business goals.

This paper is intended for those in technology roles, such as chief technology officers (CTOs), architects, developers, and operations team members. After reading this paper, you will understand Amazon Web Services (AWS) best practices and strategies to use when designing a cloud architectures for operations excellence. This paper does not provide implementation details or architectural patterns. However, it does include references to appropriate resources for this information.

Operational Excellence

The operational excellence pillar includes the ability to run and monitor systems to deliver business value and to continuously improve supporting processes and procedures.

Design Principles

There are five design principles for operational excellence in the cloud:

- **Perform operations as code**: In the cloud, you can apply the same engineering discipline that you use for application code to your entire environment. You can define your entire workload (applications, infrastructure, etc.) as code and update it with code. You can script your operations procedures and automate their execution by triggering them in response to events. By performing operations as code, you limit human error and enable consistent responses to events.
- **Annotated documentation**: In an on-premises environment, documentation is created by hand, used by humans, and hard to keep in sync with the pace of change. In the cloud, you can automate the creation of annotated documentation after every build (or automatically annotate hand-crafted documentation). Annotated documentation can used by humans *and* systems. Use annotations as an input to your operations code.
- **Make frequent, small, reversible changes**: Design workloads to allow components to be updated regularly. Make changes in small increments that can be reversed if they fail (without affecting customers when possible).
- **Refine operations procedures frequently**: As you use operations procedures, look for opportunities to improve them. As you evolve your workload, evolve your procedures appropriately. Set up regular Game Days to review and validate that all procedures are effective and that teams are familiar with them.
- **Anticipate failure**: Perform “pre-mortem” exercises to identify potential sources of failure so that they can be removed or mitigated. Test your failure scenarios and validate your understanding of their impact. Test your response procedures to ensure they are effective and

that teams are familiar with their execution. Set up regular Game Days to test workload and team responses to simulated events.

- **Learn from all operational failures**: Drive improvement through lessons learned from all operational events and failures. *Share what is learned* across teams and through the entire organization.

Definition

Operational excellence in the cloud is composed of three areas:

- Prepare
- Operate
- Evolve

Operations teams need to understand the business and customer needs to effectively and efficiently support business outcomes. Operations creates and uses procedures to respond to operational events, and validates their effectiveness to support business needs. Operations collects metrics that are used to measure the achievement of desired business outcomes. Everything continues to change—your business context, business priorities, customer needs, etc. —so it's important to design operations to support evolution over time in response to change and lessons learned through their performance.

Prepare

To prepare for operational excellence you have to understand your workloads and their expected behaviors. You will then be able design them to provide insight to their status and build the procedures to support them.

To prepare for operational excellence, you need to consider the following:

- Operational priorities
- Design for operations
- Operational readiness

Operational Priorities

Your teams need to have a shared understanding of your entire workload, their role in it, and shared business goals in order to set the priorities that will enable business success. You also need to consider external regulatory and compliance requirements that may influence your priorities. Use your priorities to focus your operations improvement efforts where they will have the greatest impact (for example, developing team skills, improving workload performance, automating runbooks, or enhancing monitoring). Update your priorities as needs change.

AWS can help you educate your teams about AWS and its services to increase their understanding of how operations choices can have an impact on your workload.

You should use the resources provided by AWS Support (AWS Knowledge Center, AWS Discussion Forms, and AWS Support Center) and AWS Documentation to educate your teams. Reach out to AWS Support through AWS Support Center for help with your AWS questions.

If there are external regulatory or compliance requirements that apply to your organization, you should use the resources provided by AWS Cloud Compliance to help educate your teams so they can consider the impact on your operational priorities.

AWS Trusted Advisor is a tool that provides access to a core set of checks that recommend optimizations for your environment that may help shape your priorities. Business and Enterprise Support customers receive access to additional checks focusing on security, reliability, performance, and cost-optimization that can further help shape their priorities. Enterprise Support customers are eligible for a Cloud Operations Review designed to help them to identify gaps in their approach to operating in the cloud. They are also eligible for a Well-Architected Review of their mission-critical workloads to measure their architectures against AWS best practices. The cross-team engagement of these reviews helps to establish common understanding of your workloads and how team roles contribute to success. The needs identified through the review can help shape your priorities.

You might find that you want to emphasize a small subset of your operational priorities at some point in time. Use a balanced approach over the long term to ensure the development of needed capabilities.

Key AWS Services

The Key AWS service that supports defining your operational priorities is AWS Support. It provides a combination of tools and expertise to help you define your goals on AWS. The following services and features are also important:

- **AWS Cloud Compliance** enables you to understand the robust controls in place at AWS to maintain security and data protection in the cloud.
- **AWS Trusted Advisor** provides real-time guidance to help you provision your resources following AWS best practices.
- **Business Support** provides access to the full set of Trusted Advisor checks and guidance to provision your resources following AWS best practices.
- **Enterprise Support** customers also receive support from Technical Account Managers (TAMs) who, as designated technical points of contact, provide guidance to help you plan and build solutions using best practices, and proactively keep your AWS environment operationally healthy.

Resources

Refer to the following resources to learn more about AWS best practices for operational priorities.

Documentation

- AWS Trusted Advisor[2]
- AWS Cloud Compliance[3]
- AWS Well Architected Framework[4]
- AWS Business Support[5]
- AWS Enterprise Support[6]
- AWS Enterprise Support Entitlements[7]

- AWS Support Cloud Operations Reviews[8]
- AWS Cloud Adoption Framework[9]

Design for Operations

The design of your workload should include how it will be deployed, updated, and operated. You will want to implement engineering practices that align with defect reduction and quick and safe fixes. To understand what is happening inside your architecture, you will need to enable observation with logging, instrumentation, and insightful business and technical metrics.

In AWS, you can view your entire workload (applications, infrastructure, policy, governance, and operations) as code. It can all be defined in and updated using code. This means you can apply the same engineering discipline that you use for application code to every element of your stack.

You should use AWS CloudFormation to create version-controlled templates for your infrastructure. You should set up Continuous Integration/Continuous Deployment (CI/CD) pipelines using the AWS Developer Tools (for example, AWS CodeCommit, AWS CodeBuild, AWS CodePipeline, AWS CodeDeploy, and AWS CodeStar). Adopt practices that allow you to detect defects early, and fix or work around them safely in production. As your operations support additional workloads you can use the artifacts from AWS CloudFormation and AWS Developer Tools to share design standards. You should apply metadata using tags to enable identification of your resources for operations activities (for example, resource owner, lifecycle stage, and environment).

When operating a workload you will need logging or other instrumentation to help you know what has happened, and to be able to understand the internal state of your system. You should capture logs in Amazon CloudWatch (for example, AWS CloudTrail, AWS Lambda, Amazon VPC Flow Logs). You should inject logs from your application directly into CloudWatch using custom CloudWatch Events and the Amazon CloudWatch Logs API. For system-level logs use the CloudWatch Logs agent and the AWS CLI. You should use this log information to create a system-wide view of your operational status either using CloudWatch Dashboards or third-party tools.

To observe the internal state of the system, your application code should publish metrics using CloudWatch. Ensure that you publish business metrics as well as

technical metrics because these will help you understand your customers' behaviors. AWS enables instrumentation on distributed applications using AWS X-Ray, which provides an end-to-end view of requests as they travel through your application.

When instrumenting your workload, capture a broad set of information to enable situational awareness (for example, changes in state, user activity, privilege access, utilization counters), knowing that you can filter to select the most useful information over time. Tag your resources for organization, cost accounting, access controls, and targeting the execution of automation.

Key AWS Services

The key AWS service that supports designing for operations is Amazon CloudWatch, which allows you to monitor AWS Cloud resources and the applications that you run on AWS. The following services and features are also important:

- **AWS CloudFormation** allows you to create version-controlled standardized templates for your infrastructure.
- **AWS Developer Tools** is a set of services enabling rapid and safe delivery of software.
- **AWS X-Ray** traces user requests as they travel through your entire application, enabling analysis and debugging of distributed applications.

Resources

Refer to the following resources to learn more about AWS best practices for operations design.

Videos

- AWS re:Invent 2016: Infrastructure Continuous Delivery Using AWS CloudFormation (DEV313)[10]
- AWS re:Invent 2016: DevOps on AWS: Accelerating Software Delivery with AWS Developer Tools (DEV201)[11]
- AWS CodeStar: The Central Experience to Quickly Start Developing Applications on AWS

Documentation

- Getting Started With Amazon CloudWatch[12]
- Store and Monitor OS & Application Log Files with Amazon CloudWatch[13]
- High-Resolution Custom Metrics and Alarms for Amazon CloudWatch[14]
- Monitoring AWS Health Events with Amazon CloudWatch Events[15]
- AWS CloudFormation Documentation[16]
- AWS Developer Tools[17]
- Set Up a CI/CD Pipeline on AWS[18]
- AWS X-Ray[19]
- AWS Tagging Strategies[20]

Operational Readiness

You should use a consistent process (including checklists) to know when you are ready to go live with your workload. This will also enable you to find any areas you will need to make plans to address. You will have runbooks that document your routine activities and playbooks that guide your processes for issue resolution. You will need to have enough team members to cover operational activities (including on-call), with training on AWS, your workload, and your operations tools. You should use a governance process to make an informed decision on launching your workload.

AWS allows you to treat your operations as code, scripting your runbook and playbook activities to reduce the risk of human error. You can use resource tags with your scripts to selectively execute based on criteria you have defined (for example, environment, owner, role, or version). You can use scripted procedures to enable automation by triggering the scripts in response to events. By treating both your operations and workloads as code, you can also script and automate the evaluation of your environments.

On AWS, you can create temporary parallel environments, which lowers the risk, effort, and cost of experimentation and testing. You should test your procedures, failure scenarios, and the success of your responses in order to

identify areas you need to plan to address (for example, by holding game days and testing prior to going live).

You should script procedures on your instances using Amazon EC2 Systems Manager Run Command and use AWS Lambda to script responses to events across AWS service APIs and your own custom interfaces. Automate your responses by triggering these scripts using CloudWatch Events. You should automate workload configuration testing by making baselines using AWS Config and checking your configurations using AWS Config rules. These services will aid in determining if your workloads are aligned with best practices and standards.

Ensure that your teams have the necessary skills to successfully operate your workloads. AWS provides a number of resources including the AWS Resource Centers, AWS Blogs, and AWS Online Tech Talks, which all provide instructions, examples, and walkthroughs to educate your teams. AWS Training and Certification provides some free online training. In addition, you can register for instructor-led training to support the development of your teams' AWS skills.

Use "pre-mortems" to anticipate failure and create procedures where appropriate. When you make changes to the checklists you use to evaluate your workloads, plan what you will do with live systems that no longer comply.

Key AWS Services

The key AWS service that supports operational readiness is AWS Lambda, which enables the definition of operational procedures as code that can be triggered by events within your environment. The following services and features are also important:

- **AWS Config** allows you to track changes to your deployed CloudFormation stacks. With AWS Config rules you can evaluate whether your AWS resources comply with best practices.
- **Amazon EC2 Systems Manager** is a collection of capabilities that helps you automate management tasks on your Amazon Elastic Compute Cloud (Amazon EC2) instances.

Resources

Refer to the following resources to learn more about AWS best practices for operational readiness.

Documentation

- AWS Lambda[21]
- Amazon EC2 Systems Manager[22]
- AWS Config Rules – Dynamic Compliance Checking for Cloud Resources[23]
- How to track configuration changes to CloudFormation stacks using AWS Config[24]
- Amazon Inspector Update blog post[25]
- AWS Events and Webinars[26]
- AWS Training[27]

Operate

Operational success is measured by the outcomes and metrics you define. By understanding the operational health of your workload, you can identify when it is impacted by operational events and respond appropriately.

To operate successfully, you need to consider the following:

- Understanding operational Health
- Responding to Events

Understanding Operational Health

Your team should be able to understand the operational health of your workload easily. You will want to use metrics based on operational outcomes to gain useful insights. You should use these metrics to implement dashboards with business and technical viewpoints that will help team members make informed decisions.

AWS makes it easier to bring together and analyze your workload and operations logs so that you can know your operating status and gain insight from operations over time.

You should send log data to CloudWatch Logs and define baseline metrics to establish normal operating patterns. Create CloudWatch Dashboards that present system- and business-level views of your metrics.

You could also ingest your CloudWatch Logs log data into Amazon Elasticsearch Service (Amazon ES) and then use the built-in support for Kibana to create dashboards and visualizations of your operational health (for example, order rates, connected users, and transaction times).

In the AWS shared responsibility model, portions of monitoring are delivered to you through the AWS Service Health Dashboard (SHD) and the Personal Health Dashboard (PHD). These dashboards provide alerts and remediation guidance when AWS is experiencing events that might affect you. Customers with Business and Enterprise Support subscriptions also get access to the PHD API, enabling integration to their event management systems.

AWS also has support for third-party log analysis systems and business intelligence tools through the AWS service APIs and SDKs (for example, Grafana, Kibana, and Logstash).

Key AWS Services

The key AWS service that helps you understand operational health is Amazon CloudWatch through its feature set including metrics and dashboards. The following services and features are also important:

- **Amazon CloudWatch Logs** allows you to monitor and store logs from EC2 instances, AWS CloudTrail, and other sources.
- **Amazon ES** makes it easy to deploy, secure, operate, and scale Elasticsearch for log analytics, and application monitoring.
- **Personal Health Dashboard** provides alerts and remediation guidance when AWS is experiencing events that may impact you.
- **Service Health Dashboard** provides up-to-the-minute information on AWS service availability.

Resources

Refer to the following resources to learn more about AWS best practices for understanding operational health.

Videos

- AWS re:Invent 2015: Log, Monitor and Analyze your IT with Amazon CloudWatch (DVO315)[28]
- AWS re:Invent 2016: Amazon CloudWatch Logs and AWS Lambda: A Match Made in Heaven (DEV301)[29]

Documentation

- Store and Monitor OS & Application Log Files with Amazon CloudWatch[30]
- New – API & CloudFormation Support for Amazon CloudWatch Dashboards[31]
- AWS Answers: Centralized Logging[32]

Responding to Events

You should anticipate operational events, both planned (for example, sales promotions, deployments, and failure tests) and unplanned (for example, surges in utilization and component failures). You should use your existing runbooks and playbooks to deliver consistent results when you respond to alerts. Defined alerts should be owned by a role or a team that is accountable for the response and escalations. You will also want to know the business impact of your system components and use this to target efforts when needed. You should perform a root cause analysis (RCA) after events, and then prevent recurrence of failures or document workarounds.

AWS simplifies your event response by providing tools supporting all aspects of your workload and operations as code. These tools allow you to script responses to operations events and trigger their execution in response to monitoring data. In AWS, you can improve recovery time by replacing failed components with known good versions, rather than trying to repair them. You can then carry out analysis on the failed resource out of band.

There are multiple ways to automate the execution of runbook and playbook actions on AWS. To respond to an event from a state change in your AWS resources, or your own custom events, you should create CloudWatch rules to trigger responses through CloudWatch targets (for example, Lambda functions, Amazon Simple Notification Service (Amazon SNS) topics, and Amazon EC2 Container Service (Amazon ECS) tasks). To respond to a metric that crosses a threshold for a resource (for example, wait time), you should create CloudWatch alarms to perform one or more actions using Amazon EC2 actions, Auto Scaling actions, or to send a notification to an Amazon SNS topic. If you need to perform custom actions in response to an alarm, invoke Lambda through Amazon SNS notification. Use Amazon SNS to publish event notifications and escalation messages to keep people informed.

AWS also supports third-party systems through the AWS service APIs and SDKs. There are a number of tools provided by partners and third-parties that allow for monitoring, notifications, and responses. Some of these tools are New Relic, Splunk, Loggly, SumoLogic, and Datadog.

Know when a human decision is needed before an action is taken. When possible, have that decision made before the action is required. You should keep critical manual procedures available for use when automated procedures fail.

Key AWS Services

The key AWS service that helps you automate response to events is AWS Lambda, which enables the definition of operational procedures as code that can be triggered by events within your environment. The following services and features are also important:

The following services and features are also important:

- **Amazon CloudWatch** is used for the collection of logs and metrics. It enables the triggered execution of responses.
- **Amazon CloudWatch Events** delivers a near real-time stream of system events that can be matched to rules you define to trigger automated responses.
- **Amazon SNS** is a flexible, fully managed publication subscription messaging and mobile notifications service for coordinating the delivery

of messages to subscribing endpoints and clients. It enables you invoke Lambda functions in response to alarms.

- **Auto Scaling** helps you maintain application availability and allows you to dynamically scale your Amazon EC2 capacity up or down automatically according to conditions you define.
- **Amazon EC2 Systems Manager** is a collection of capabilities that helps you automate management tasks on your EC2 instances.

Resources

Refer to the following resources to learn more about AWS best practices for responding to events.

Video

- AWS re:Invent 2016: Automating Security Event Response, from Idea to Code to Execution (SEC313)[33]

Documentation

- What is Amazon CloudWatch Events?[34]
- How to Automatically Tag Amazon EC2 Resources in Response to API Events[35]
- EC2 Run Command is Now a CloudWatch Events Target[36]
- New – High-Resolution Custom Metrics and Alarms for Amazon CloudWatch[37]

Evolve

Evolution is the continuous cycle of improvement over time. Implement frequent small incremental changes based on the lessons learned from your operations activities.

To evolve your operations over time, you need to consider the following:

- Learning from experience
- Sharing learnings

Learning from Experience

When things fail you will want to ensure that your team, as well as your larger engineering community, learns from those failures. You should analyze failures to identify lessons learned and plan improvements. You will want to regularly review your lessons learned with other teams to validate your insights. It's essential that you regularly provide time for analysis, experimentation, and making improvements.

On AWS, you can easily aggregate the logs of all your operations activities, workloads, and infrastructure to create a detailed activity history. You can then use AWS tools to analyze your operations over time (for example, identify trends, correlate events and activities to outcomes, and compare and contrast between environments and across systems) to reveal opportunities for improvement.

On AWS, you can create temporary duplicates of environments, lowering the risk, effort, and cost of experimentation and testing. These duplicated environments can be used to test the conclusions from your analysis, experiment, and develop and test planned improvements.

You should use CloudTrail to track API activity (through the AWS Management Console, CLI, SDKs, and APIs) to know what is happening across your accounts. Track your AWS Developer Tools deployment activities with CloudTrail and CloudWatch. This will add a detailed activity history of your deployments and their outcomes to your CloudWatch Logs log data.

You should also ingest your CloudWatch Logs log data into Amazon ES and then use the built-in support for Kibana to create visualizations and perform retrospective analysis of your operations over time.

Alternatively, you could export your CloudWatch data to Amazon Simple Storage Service (Amazon S3), analyze it with Amazon Athena, and use Amazon QuickSight to perform analysis, create visualizations, and gain insights.

AWS also has support for third- party log analysis systems and business intelligence tools through the AWS service APIs and SDKs (for example, Grafana, Kibana, and Logstash).

Perform cross-team reviews by engaging business, developer, and other operations teams to discuss and validate insights and identify areas for improvement. Look for opportunities to improve in all of your environments (for example, development, test, and production).

Key AWS Services

The key AWS service that helps you learn from experience is Amazon ES, which allows you to go analyze your log data to gain actionable insights quickly and securely. The following services and features are also important:

- **Amazon QuickSight** is a business analytics service that makes it easy to build visualizations, perform ad-hoc analysis, and quickly get insights from your data.
- **Amazon Athena** is a serverless interactive query service that makes it easy to analyze data in Amazon S3.
- **Amazon S3** can be used for collection and archival retention of logs.
- **Amazon CloudWatch** is used for the collection of logs and metrics and the creation of dashboards.

Resources

Refer to the following resources to learn more about AWS best practices for learning from experience.

Video

- Serverless Big Data Analytics - Amazon Athena & Amazon QuickSight[38]

Documentation

- View AWS CodeDeploy logs in Amazon CloudWatch console[39]
- Analyzing VPC Flow Logs with Amazon Kinesis Firehose, Amazon Athena, and Amazon QuickSight[40]

Share Learnings

You should share what your teams learn to increase the benefit across your organization. You will want to share information and resources to prevent

avoidable errors and ease development efforts. This will allow you to focus on delivering features.

On AWS, application, compute, infrastructure, and operations can be defined and managed using code methodologies. This allows for easy release, sharing, and adoption. Many AWS services and resources are designed to be shared across accounts, enabling you to share products and learnings across your teams.

Use AWS Identity and Access Management (IAM) to define permissions enabling controlled access to the resources you wish to share within and across accounts. You should then use version-controlled AWS CodeCommit repositories to share application libraries, scripted procedures, procedure documentation, and other system documentation. Share your compute standards by sharing access to your AMIs and by authorizing the use of your Lambda functions across accounts. You should also share your infrastructure standards as CloudFormation templates. When you publish new resources or updates, use Amazon SNS to publish notifications. Subscribers can use Lambda to get new versions.

Through the AWS APIs and SDKs, you can integrate external and third-party tools and repositories (for example, GitHub, BitBucket, and SourceForge).

When sharing what you have learned and developed, be careful to structure permissions to ensure the integrity of shared repositories.

Key AWS Services

The key AWS service that supports sharing learnings is AWS IAM, which enables you to manage the sharing of resources within and across accounts. The following services and features are also important:

- **Amazon SNS** enables event-based notification of publishing of resources to subscribers.
- **AWS CodeCommit** provides a version-controlled repository for your operations as code that can be shared through IAM.
- **AWS Lambda** enables the definition of operational procedures as code that can be shared across accounts.

- **AWS CloudFormation** allows you to create version-controlled standardized templates for your infrastructure.
- **Amazon Machine Images (AMIs)** are predefined operating system templates for your EC2 compute environments.

Resources

Refer to the following resources to learn more about AWS best practices for sharing learnings.

Video

- AWS re:Invent 2014 | (SEC302) Delegating Access to Your AWS Environment[41]

Documentation

- Share an AWS CodeCommit Repository[42]
- Easy Authorization of AWS Lambda Functions[43]
- Sharing an AMI with Specific AWS Accounts[44]
- Speed Template Sharing with an AWS CloudFormation Designer URL[45]
- Using AWS Lambda with Amazon SNS[46]

Conclusion

Operational excellence is an ongoing effort. Every operational event and failure should be treated as an opportunity to improve the operations of your architecture. By understanding the needs of your workloads, predefining runbooks for routine activities, and playbooks to guide issue resolution, using the operations as code features in AWS, and maintaining situational awareness, your operations will be ready and responsive when events occur. Through focusing on incremental improvement based on operational priorities, and lessons learned from event response and retrospective analysis, you will enable the success of your business by increasing the efficiency and effectiveness of your operations.

AWS strives to help you build and operate architectures that maximize efficiency while you build highly responsive and adaptive deployments. To make

your workloads truly operationally excellent, you should use the best practices discussed in this paper.

Contributors

- Philip Fitzsimons, Sr. Manager Well-Architected, Amazon Web Services
- Brian Carlson, Operations Lead Well-Architected, Amazon Web Services
- Jon Steele, Sr. Technical Account Manager, Amazon Web Services
- Ryan King, Technical Program Manager, Amazon Web Services

Further Reading

For additional help, please consult the following sources:

- AWS Well-Architected Framework[47]

Notes

[1] https://aws.amazon.com/well-architected

[2] https://aws.amazon.com/premiumsupport/trustedadvisor/

[3] https://aws.amazon.com/compliance/

[4] https://aws.amazon.com/architecture/well-architected/

[5] https://aws.amazon.com/premiumsupport/business-support/

[6] https://aws.amazon.com/premiumsupport/enterprise-support/

[7] https://aws.amazon.com/blogs/aws/aws-enterprise-support-update-training-credits-operations-review-well-architected/

[8] https://aws.amazon.com/about-aws/whats-new/2016/04/aws-support-introduces-operations-reviews-recommendations-and-reporting-available-through-enterprise-support-plan/

[9] https://aws.amazon.com/professional-services/CAF/

[10] https://www.youtube.com/watch?v=TDalsML3QqY

[11] https://www.youtube.com/watch?v=-ddpq2VQNxo

[12] https://aws.amazon.com/cloudwatch/getting-started/

[13] https://aws.amazon.com/blogs/aws/cloudwatch-log-service/

[14] https://aws.amazon.com/blogs/aws/new-high-resolution-custom-metrics-and-alarms-for-amazon-cloudwatch/

[15] http://docs.aws.amazon.com/health/latest/ug/cloudwatch-events-health.html

[16] http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html

[17] https://aws.amazon.com/products/developer-tools/

[18] https://aws.amazon.com/getting-started/projects/set-up-ci-cd-pipeline/

[19] https://aws.amazon.com/xray/

[20] https://aws.amazon.com/answers/account-management/aws-tagging-strategies/

[21] https://aws.amazon.com/lambda/

[22] https://aws.amazon.com/ec2/systems-manager/

[23] https://aws.amazon.com/blogs/aws/aws-config-rules-dynamic-compliance-checking-for-cloud-resources/

[24] https://aws.amazon.com/blogs/mt/how-to-track-configuration-changes-to-cloudformation-stacks-using-aws-config/

[25] https://aws.amazon.com/blogs/aws/category/amazon-inspector/

[26] https://aws.amazon.com/about-aws/events/

[27] https://aws.amazon.com/training/

[28] https://www.youtube.com/watch?v=ZaOR-ybLJF0&t=1232s

[29] https://www.youtube.com/watch?v=xaFaVeoA9V8

[30] https://aws.amazon.com/blogs/aws/cloudwatch-log-service/

[31] https://aws.amazon.com/blogs/aws/new-api-cloudformation-support-for-amazon-cloudwatch-dashboards/

[32] https://aws.amazon.com/answers/logging/centralized-logging/

[33] https://www.youtube.com/watch?v=x4GkAGe65vE

[34] http://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html

[35] https://aws.amazon.com/blogs/security/how-to-automatically-tag-amazon-ec2-resources-in-response-to-api-events/

[36] https://aws.amazon.com/blogs/aws/ec2-run-command-is-now-a-cloudwatch-events-target/

[37] https://aws.amazon.com/blogs/aws/category/amazon-cloud-watch/

[38] https://www.youtube.com/watch?v=m1HTL_SJHrE

[39] https://aws.amazon.com/blogs/devops/view-aws-codedeploy-logs-in-amazon-cloudwatch-console/

[40] https://aws.amazon.com/blogs/big-data/analyzing-vpc-flow-logs-with-amazon-kinesis-firehose-amazon-athena-and-amazon-quicksight/

[41] https://www.youtube.com/watch?v=0zJuULHFS6A&t=849s

[42] http://docs.aws.amazon.com/codecommit/latest/userguide/how-to-share-repository.html

[43] https://aws.amazon.com/blogs/compute/easy-authorization-of-aws-lambda-functions/

[44] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sharingamis-explicit.html

[45] https://aws.amazon.com/blogs/devops/speed-template-sharing-with-an-aws-cloudformation-designer-url/

[46] http://docs.aws.amazon.com/lambda/latest/dg/with-sns-example.html

[47] https://aws.amazon.com/well-architected