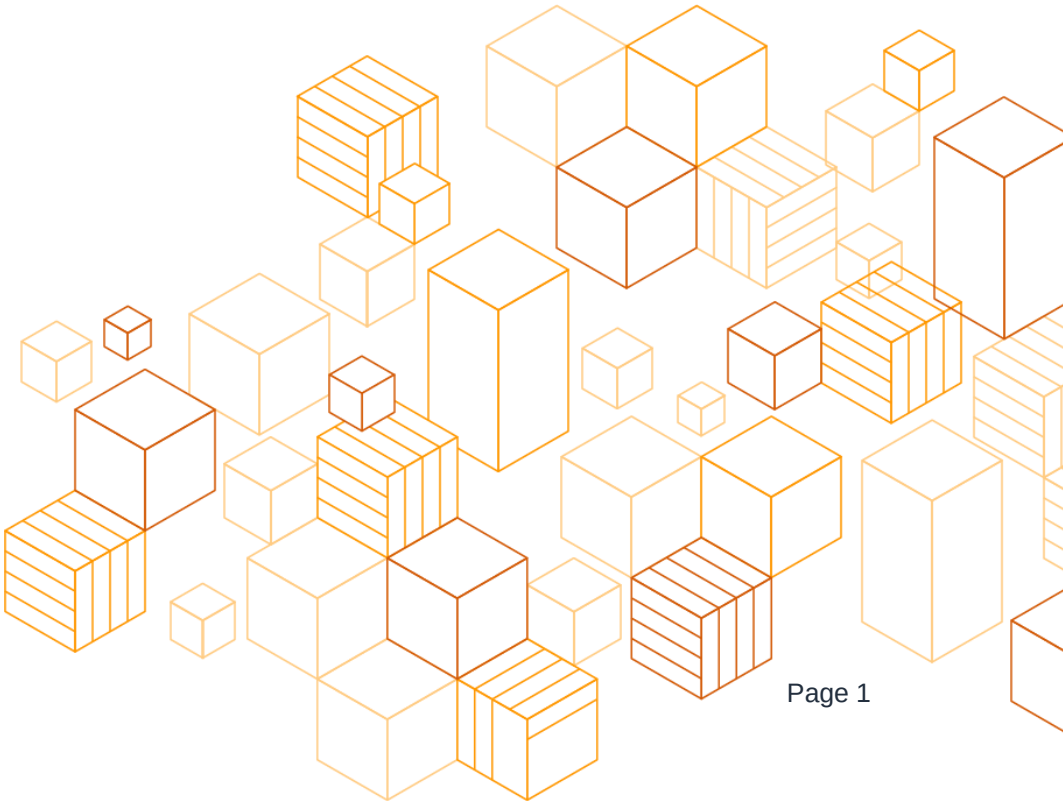
JPX arrownet Connection for AWS

Implementation Guide

May 2020



Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. 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

Overview

This Implementation Guide focuses on creating a connection with arrownet (arrownet version 2) operated by Japan Exchange Group, Inc. (JPX) with Amazon Web Services (AWS). This guide presents example scenarios and reference architecture diagrams for connecting with arrownet. Each reference architecture is explained using terms and definitions from the AWS Well-Architected Framework .

This guide is intended for IT decision-makers and infrastructure/networking professionals who are familiar with the basic concepts of networking, operating systems, and operations management.

AWS Cloud Architecture for JPX arrownet Connection

This section describes the necessary architecture for establishing a connection with JPX arrownet using the AWS Cloud.

This implementation guide explains reference architectures based on example scenarios. Follow the scenario that best suits your needs. In order to maximize the security and efficiency benefits of the AWS Cloud, you must ensure that you follow the Well-Architecture Framework through construction phases (requirements definition, design, construction, and testing).

Review the following statements before you begin. For more information, check the JPX arrownet guidelines.

- Confirm the JPX supported services before using an AWS Access Point.
- When using an AWS Access Point, JPX recommends setting up a redundant configuration using an arrownet version 2.0 line in addition to having a cloud connection. This guide describes an AWS network architecture—this architecture alone cannot achieve the redundant configuration recommended by JPX. Therefore, you must design a network connection to arrownet while taking into consideration the existing connection service (arrownet version 2.0 line) provided by arrownet. This will guarantee the JPX recommended availability.
- Only unicast is available for AWS connection in arrownet version 2.0. Multicast is not available.
- AWS connection in arrownet version 2.0 can be used only on the JPX primary site. It cannot be used when switching sites.

Connection Scenarios with arrownet

The AWS Cloud connects to JPX arrownet through the AWS Direct Connect Private Virtual Interface (Private VIF).

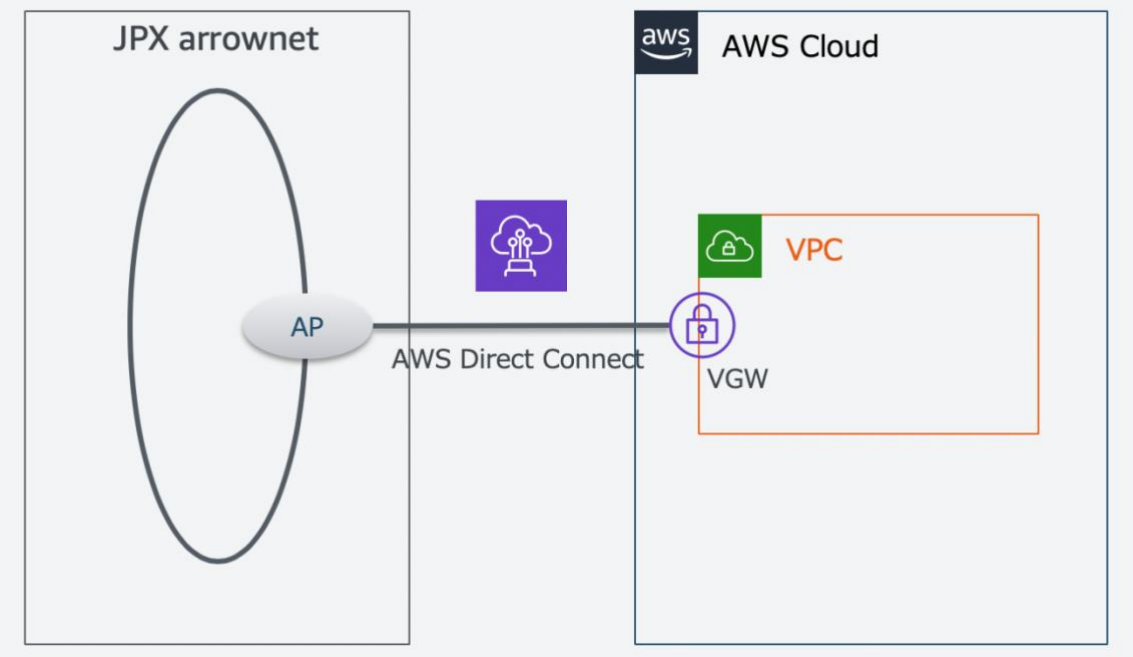


Figure 1 – Connection basics between arrownet and the AWS Cloud

You must apply to JPX for permissions to connect through a Private VIF. For details on the application procedure, refer to the JPX arrownet version2.0 guidelines.

The following table contains example connection scenarios.

Table 1 – Connection scenarios

Scenario Classification	Overview	Detailed Explanation
Scenario 1	Confining and Managing Traffic within the AWS Cloud	Confines and handles traffic from JPX in AWS that is available to market participants. This scenario is designed for market participants who have already migrated their workloads to AWS, as well as for market participants (for example, Fintech Startups) that are newly implementing workloads on AWS.
Scenario 2	Forwarding Traffic to on-premises Data Centers	Forwards traffic to on-premises data centers used by market participants, where traffic from JPX can be processed. This scenario is designed for market participants who use AWS as a transit center and whose workload communication with JPX is kept in on-premises data centers.
Scenario 3	Log Aggregation and Analysis Architecture	An architecture to aggregate and analyze logs generated from the traffic between JPX and the AWS Cloud.

Scenario 1. Confining and Managing Traffic within the AWS Cloud

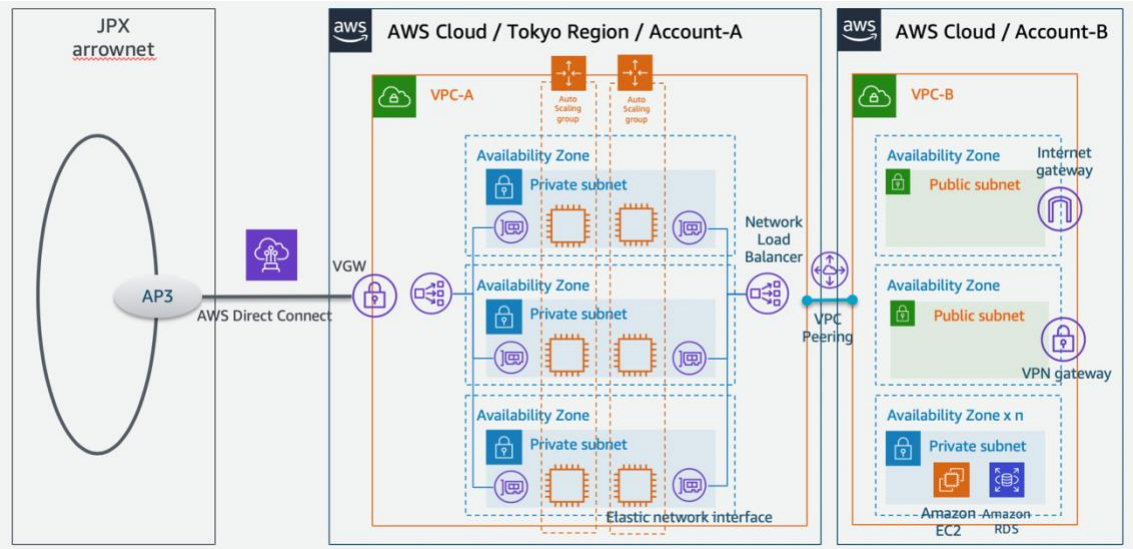


Figure 2 – Reference Architecture for Confining and Managing Traffic within the AWS Cloud

Design Principles

Select an architecture that does not allow internal communication of market participants to flow into arrownet, and where market participants do not need to interact with JPX.

The Virtual Private Cloud (VPC) is divided into VPC-A and VPC-B. The architecture is designed so that VPC-A is dedicated to handle traffic with JPX, and resources for market participants' workloads are allocated to VPC-B.

In the unlikely event that unexpected traffic flows from VPC-B of a market participant to VPC-A, the proxy server function of VPC-A (or the security group function) can prevent traffic from flowing to arrownet. Conversely, VPC-A can also prevent unexpected traffic from the arrownet end.

Do not place the Internet Gateway or VPN Gateway used by market participants in VPC-A, which contains the Private VIF of AWS Direct Connect provided by arrownet.

When communicating with a VPC-B owned by market participants, you can use VPC Peering between VPC-A and VPC-B, AWS Transit Gateway, or AWS PrivateLink.

Architecture Explanation

> **Important:** You must use the AWS Tokyo Region for this scenario. Using the AWS Tokyo Region is a compliance requirement defined by the Financial Instruments and Exchange Act.

The following steps describe this scenario's reference architecture.

1. Set up VPC-A in the Tokyo Region and attach VIF to the Virtual Private Gateway (VGW).
2. Deploy a Network Load Balancer (NLB) and create a proxy server using Amazon Elastic Compute Cloud (EC2) virtual server, placing it in the 3 Availability Zones (AZ) of ap-northeast-1a, ap-northeast-1c, and ap-northeast-1d.
3. Adjust the Auto Scaling settings to Max: 3, Min: 3, Desirable: 3. If the protocol used to connect to the information services of JPX is limited to http or https, Application Load Balancer (ALB) can be used instead of NLB. In addition, the server or browser of market participants who access NLB or ALB must be set up so that it can resolve names via a DNS server.

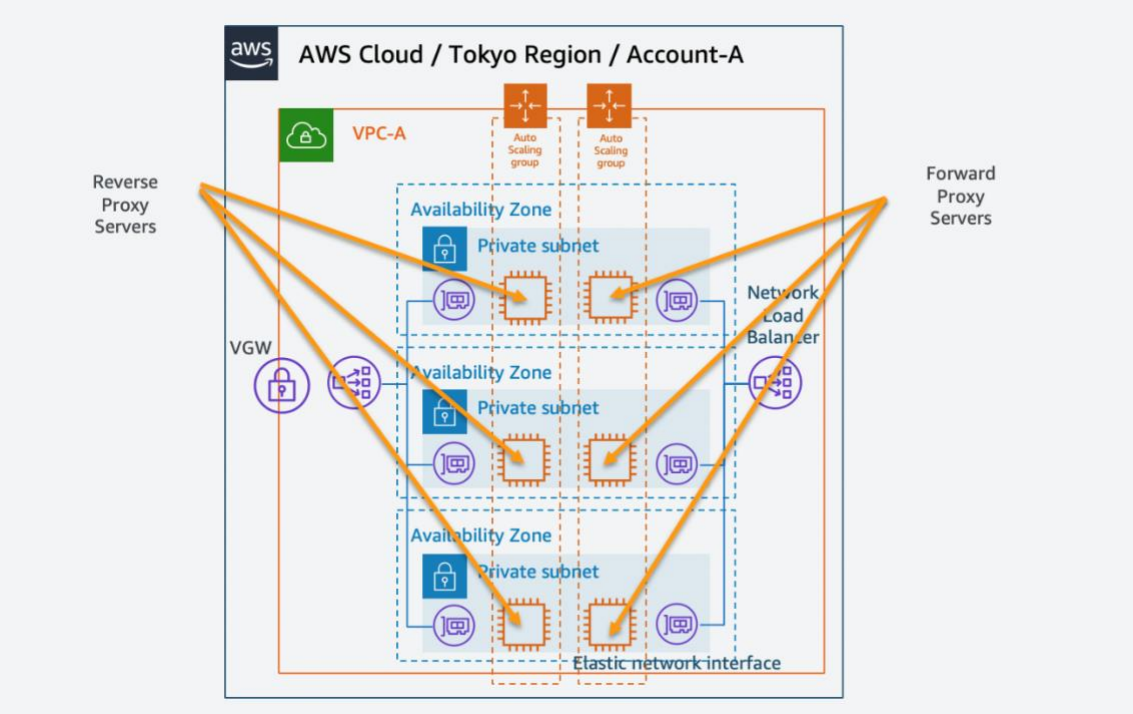


Figure 3 – Proxy server settings

Refer to the following detailed information about EC2 settings:

1. Since the network bandwidth changes depending on the instance type, select the optimal instance type while measuring the traffic volume with JPX. Depending on the instance type, the network bandwidth may not be clearly defined. Keep this in mind when selecting the instance type.
2. When using the T2/T3 instance type, select the **T2/T3 Unlimited** option when launching the instance. This is a setting to avoid traffic errors due to the depletion of CPU credits.
3. We recommend that you select the **enable Amazon CloudWatch detailed monitoring** option to monitor the operating status of EC2. This guide recommends the use of a 3AZ Multi-AZ configuration, or a minimal 2 Availability Zone (AZ) configuration at the very least. We do not recommend the 1Availability Zone (AZ) configuration (Single-AZ configuration) from a reliability standpoint based on the Well-Architected Framework. Auto Scaling is used for the purpose of automatic recovery in case of instance failure called Auto Healing, and this setting maintains a startup state of 2-3 EC2 instances at all times. In order to make EC2 a proxy server, install the optimal software that complies to each market participant's security policy. There are two sets of NLB + EC2 + Auto Scaling combinations. One is for the forward proxy for traffic to JPX, and the other would be for the reverse proxy for traffic from JPX. These proxies are configured on EC2 and must be started up reliably in order to maintain communications with arrownet. As such, adopt Reserved Instances (RI) by the AZ. Adopting RI also improves cost efficiency. Market participants can purchase RI to meet their own investment plans. On the other hand, because the securities market experiences large traffic fluctuations, we recommend testing for multiple instance types during the initial construction stage to prepare for any necessary instance type changes.
4. Set the VPC-B to **Tokyo Region** and allow communications with VPC-A. Deploy the necessary resources to handle the market participant's workload.

Scenario 2. Forwarding Traffic to On-Premises Data Centers

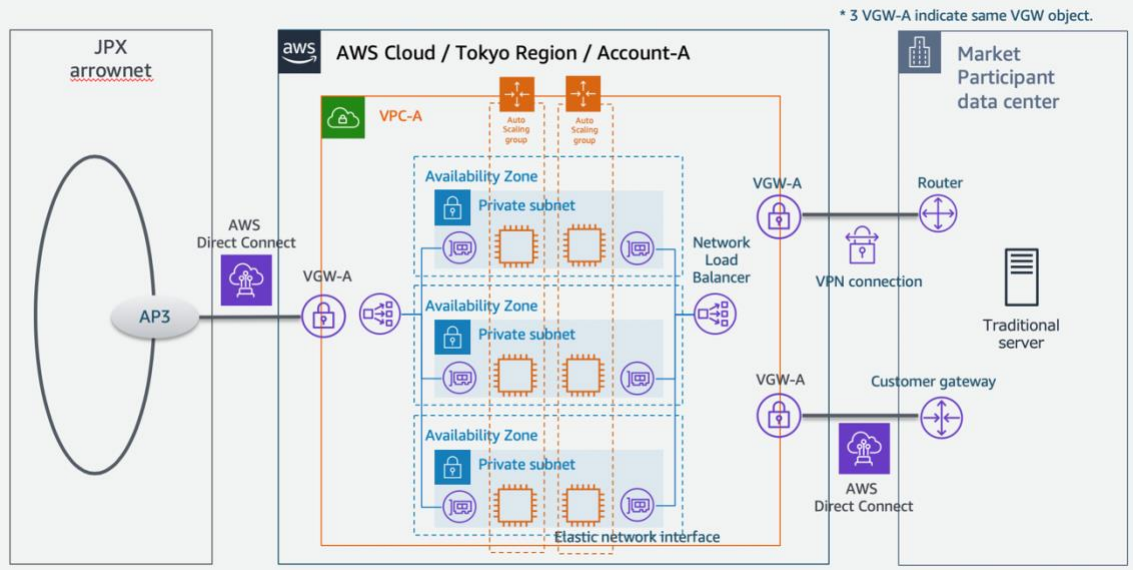


Figure 4 –Reference Architecture for Forwarding Traffic to On-Premises Data Centers

Design Principles

As with in Scenario 1, market participants are required to have an architecture that prevents any of their internal communications (between those who are not essential to JPX interactions) from flowing into arrownet.

- Use VPC-A exclusively to handle JPX traffic. This architecture forwards traffic to the market participant's on-premises data centers through VPC-A, using AWS Direct Connect or AWS Site-to-Site VPN.
- The market participant's Internet Gateway or VPN Gateway will not be assigned to VPC-A, as it receives a Private VIF from AWS Direct Connect (provided by arrownet).

Architecture Explanation

Important: You must use the AWS Tokyo Region for this scenario. Using the AWS Tokyo Region is a compliance requirement defined by the Financial Instruments and Exchange Act.

Using AWS Transit Gateway, you can transfer traffic directly to the on-premises data center via AWS Direct Connect, or AWS Site-to-Site VPN, without building a VPC-A. In this scenario, there is a risk that traffic not permitted by JPX might flow to the arrownet side if the Route Table settings on AWS Transit Gateway are set incorrectly. In order to prevent this incorrect flow, build a VPC-A and implement traffic control by using security groups for VPC-A. *This is the only difference in the architecture in Scenario 2 as compared to the architecture described in Scenario 1.*

Scenario 3. Log Aggregation and Analysis Platform

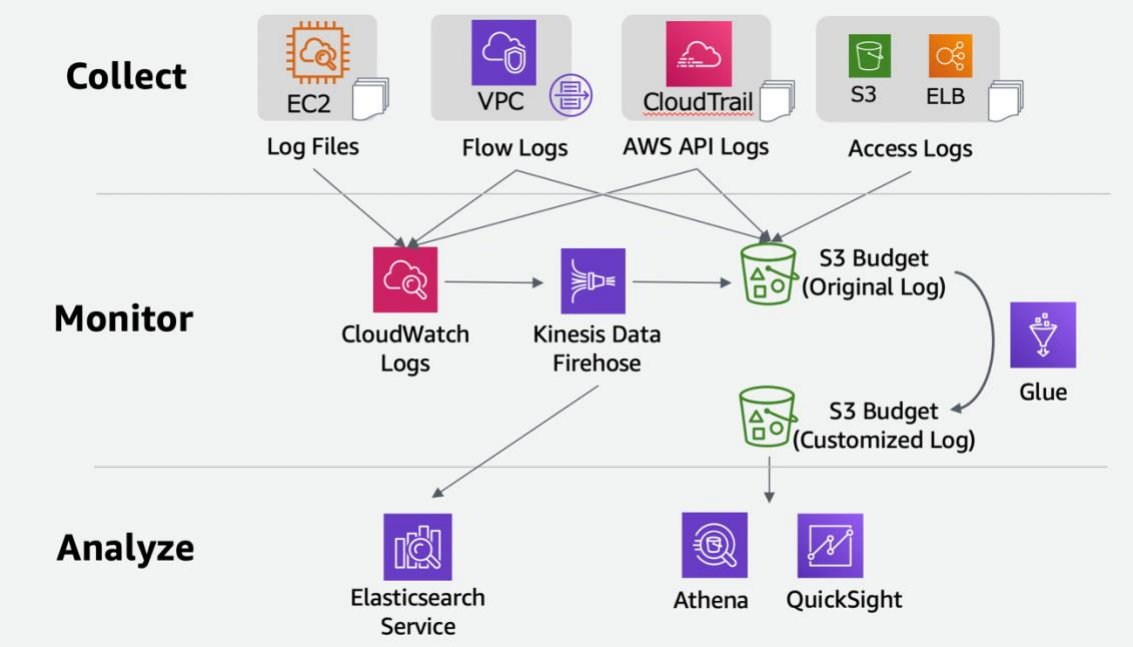


Figure 5 – Log Aggregation and Analysis Platform

Design Principles

We recommend preparing systems to aggregate and analyze logs for internal audits, and to respond to incidents. Figure 5 above depicts a general-purpose architecture used by AWS for analyzing logs. Market participants can use their trusted services or any other tools that meet their own security rules—for example, using ETL (extract, transform, load) instead of AWS Glue or BI (business intelligence) instead of Amazon QuickSight.

- This architecture aggregates the AWS Application Programming Interface (API) logs in S3. These logs are generated by essential services in Scenario 1 or Scenario 2 and record their operational results.

- Prepare a function in advance so that you have the ability to analyze the aggregated logs in S3 whenever it is necessary.
- Always retain a copy of the original log when customizing new or additional logs.

Architecture Explanation

For Scenario 1 and Scenario 2, the proxy server logs, VPC logs, AWS product settings, AWS API execution logs (generated when changes happen), and access records to load balancers such as NLB/ALB and S3 are essential for managing internal audits or responding to incidents. You should create a system that aggregates these logs in the S3 bucket.

In the initial stages, logs recorded by Amazon CloudWatch Logs as part of the AWS monitoring function are aggregated in real time into S3 buckets through Amazon Kinesis Data Firehose. Alternatively, if your data can be batch-processed at approximately 12-hour intervals, you can use the CloudWatch Logs feature to export your data to S3. Or, use AWS CloudTrail and have your run results saved directly into S3.

AWS Glue periodically converts logs in the original S3 bucket to a csv or columnar format and customizes it in terms of size and format so it is suitable for data processing and data analysis. It then saves this data into a separate S3 bucket meant for customized data. Prep your data on Amazon QuickSight via Amazon Athena ahead of time so you are able to perform data analysis as and when necessary. Using a combination of Amazon Athena and Amazon QuickSight keeps these operations serverless. It is also a great option in terms of cost performance. You can also change this with the Amazon Elasticsearch Service or other third-party data analysis services.

We recommend retaining the original logs if you are using AWS Glue for customization. in order to prevent any possible loss of data that could arise from data customizations. To keep logs for a longer period of time, you can adopt mechanisms to automatically archive old data to Amazon S3 Glacier, which offers lower pricing for long-term storage.

It is up to market participants to decide the extent to which they will implement these systems. However, the minimum threshold we recommend is to centralize and aggregate data in S3 buckets in advance. It is impossible to investigate logs have not been aggregated.

AWS Well-Architected Framework

The AWS Well-Architected Framework helps cloud architects build secure, high-performance, resilient, and efficient infrastructures. The framework is based on five pillars (operational excellence, security, reliability, performance efficiency, and cost optimization) and provides a consistent approach for you and your partners to evaluate your architecture and to implement designs that you can scale over time.

The Five Pillars of the Well-Architected Framework

Operational Excellence

The operational excellence pillar focuses on running and monitoring systems to deliver business value, and continually improving processes and procedures. Key topics include managing and automating changes, responding to events, and defining standards to successfully manage daily operations.

Security

The security pillar focuses on protecting information and systems. Key topics include confidentiality and integrity of data, identifying and managing who can do what with privilege management, protecting systems, and establishing controls to detect security events.

Reliability

The reliability pillar focuses on the ability to prevent, and quickly recover from failures to meet business and customer demand. Key topics include foundational elements around setup, cross project requirements, recovery planning, and how we handle change.

Performance Efficiency

The performance efficiency pillar focuses on using IT and computing resources efficiently. Key topics include selecting the right resource types and sizes based on workload requirements, monitoring performance, and making informed decisions to maintain efficiency as business needs evolve.

Cost Optimization

Cost Optimization focuses on avoiding unnecessary costs. Key topics include understanding and controlling where money is being spent, selecting the most appropriate and right number of resource types, analyzing spend over time, and scaling to meet business needs without overspending.

Considerations for Operational Excellence

The following table describes a design guide based on the operational excellence considerations from the Well-Architected Framework.

Table 2 – Considerations for Operational Excellence

Considerations	Questions	Guide
OPS1	**How do you determine what your priorities are?**	• **Evaluate compliance requirements** Use AWS's Tokyo Region to comply with the compliance requirements set out within the Financial Instruments and Exchange Act. • **Evaluate threat landscape** Separate the VPC (Virtual Private Cloud) to prevent the market participants' internal communication from flowing into JPX. • **Evaluate trade-offs** This guide emphasizes safe use at optimal costs. Consider the market participants' needs while in the design and testing phases in order to achieve an optimal balance.
OPS2	**How do you design your workload so that you can understand its state?**	• **Implement and configure cloud telemetry** Configure it such that you can monitor the operational status of the NLB and EC2 through Amazon CloudWatch. Consider services and software metrics that come from third-party services (Amazon CloudWatch cannot retrieve that data).

Considerations	Questions	Guide
OPS3	**How do you reduce defects, ease remediation, and improve flow into production?**	• **Test and validate changes** • **Use configuration management systems** • **Use build and deployment management systems** • **Share design standards** • **Implement practices to improve code quality** • **Make frequent, small, reversible changes** • **Fully automate integration and deployment** We recommend applying the concept of "Infrastructure as Code" and automating deployment using AWS Cloud Development Kit (CDK) or AWS CloudFormation. Using CDK and AWS CloudFormation enables you to define information on the environment settings using code. You can use GitHub or AWS CodeCommit to manage configurations. You can also use AWS CloudFormation to manage deployments. Using code enables organizations to establish protocol and templates, which then allow them to standardize their approach and share best practices for building environments with code. Any post-production changes can be achieved by applying small code modifications in an incremental manner. These modifications are deployed in a test environment for verification and validation. Human errors can occur when you create an environment manually using AWS Management Console. Constructing the environment with code reduces room for human errors. The use of code increases the team's overall productivity and reduces reliance on the skill and knowledge levels of individual team members.

Considerations	Questions	Guide
OPS4	**How do you mitigate deployment risks?**	• **Plan for unsuccessful changes** • **Test and validate changes** • **Use deployment management systems** • **Test using limited deployments** • **Deploy using parallel environments** • **Deploy frequent, small, reversible changes** • **Fully automate integration and deployment** • **Automate testing and rollback** Refer to [OPS3] for more details.
OPS5	**How do you know that you are ready to support a workload?**	• **Ensure personnel capability** Use arrownet's test systems and ensure that your development and operational teams undergo sufficient tests/trainings. It is especially important to verify the usefulness of the runbooks and playbooks listed below through tests/trainings. • **Use runbooks to perform procedures** Runbooks are documented procedures to achieve specific outcomes. We recommend preparing procedures in advance so you can manage changes, test events from JPX, and tests from market participants. • **Use playbooks to identify issues** Playbooks are documented processes to investigate issues. We recommend consolidating procedures in advance for issues that you can anticipate. This enables you to prepare for trouble arising from periodic AWS maintenance or from unexpected arrownet issues. Validate playbook effectiveness as part of the trainings mentioned above.

Considerations	Questions	Guide
OPS6	**How do you understand the health of your workload?**	• **Identify Key Performance Indicators (KPIs)** • **Define workload metrics** • **Collect and analyze workload metrics** • **Establish workload metrics baselines** • **Learn expected patterns of activity for workloads** Collect logs from applications that handle communications with JPX. Assess the response time and traffic over specific time intervals. Use this metric to help you understand workload health. Alternatively, collect statistical data for these metrics while still in the testing stages after connecting to JPX. Measure baselines during regular conditions ahead of time, and get a grasp of how these metrics will trend over a day of operations. • **Alert when workload outcomes are at risk** • **Alert when workload anomalies are detected** Use Amazon Simple Notification Service (SNS) to raise alerts when performance deviates from the metrics baseline that has been measured in advance. Some third-party software/services and Operational Support Systems (OSS) also support alerts that are linked to metrics and events. Consider how you can work with the market participant's operation monitoring systems. • **Validate the achievement of outcomes and the effectiveness of KPIs and metrics** Periodically assess the effectiveness of your series of operations, such as the collection and analysis of metrics, and anomaly alerts. You could use monthly meetings to conduct these assessments.

Considerations	Questions	Guide
OPS7	**How do you understand the health of your operations?**	• **Identify KPIs** • **Define operations metrics** • **Collect and analyze operations metrics** • **Establish operations metrics baselines** • **Learn expected patterns of activity for operations** Verify that operations are performing as expected by checking the outcomes of temporary changes in settings during tests and by checking operations outcomes during deployment. Use Amazon CloudWatch to verify logs. Use AWS Config to verify change outcomes. Use AWS CloudTrail to verify API execution logs from AWS. We recommend storing these results so you can compare current results against past results to verify that your operations have executed correctly. • **Alert when operation outcomes are at risk** • **Alert when operation anomalies are detected** We recommend raising an alert whenever undesired operations are executed. AWS Config Rules and AWS CloudTrail have functions to raise alerts whenever undesired changes are made. Some examples include situations where an Internet Gateway is attached to a VPC that is communicating with arrownet or when changes are applied to the settings of a security group. • **Validate the achievement of outcomes and the effectiveness of KPIs and metrics** After resolving issues and testing events, provide an analysis of the results and recommend remedial action to stakeholders based on metrics.

Considerations	Questions	Guide
OPS8	**How do you manage workload and operations events?**	• **Use processes for event, incident, and problem management** Develop playbooks (mentioned in [OPS5]) to prepare processes that address events, incidents, or problems. • **Use a process for root cause analysis** Perform a root cause analysis when unexpected issues occur and add the mitigations to playbook. • **Have a process for each alert** • **Prioritize operational events based on business impact** • **Define escalation paths** • **Enable push notifications** Consider the market participant's escalation rules and discuss in advance a system that would automatically notify them of maintenance or problems. • **Communicate status through dashboards** We recommend creating a mechanism that allows stakeholders to understand the current status during events. At AWS, we keep track of our metrics status by using Amazon CloudWatch dashboards. It is also possible to provide custom metrics. Alternatively, use Amazon ElasticSearch Service or Amazon QuickSight to create dashboards that are easier to digest visually. • **Automate responses to events** You can automate responses to events if a prescribed set of corresponding processes already exists on playbook. Consider using an alert-linked automatic recovery mechanism using AWS Lambda or AWS Auto Scaling.

Considerations	Questions	Guide
OPS9	**How do you evolve operations?**	• **Have a process for continuous improvement** • **Implement feedback loops** • **Define drivers for improvement** • **Validate insights** • **Perform operations metrics reviews** • **Document and share lessons learned** • **Allocate time to make improvements** We recommend setting up an organizational structure to make regular operational improvements in line with the content described in [OPS1—8].

Considerations for Security

The following table describes a design guide based on Security considerations from the Well-Architected Framework.

Table 3 – Considerations for "Security"

Considerations	Questions	Guide
SEC1	**How do you manage credentials and authentication?**	• **Define Identity and Access Management (IAM) requirements** Establish IAM requirements that meet the security rules defined by the market participant's organization. • **Secure AWS root user** As the root user has strong privileges, we recommend that you set up Multi-Factor Authentication (MFA) to secure this user and limit his or her use. • **Enforce use of MFA** We recommend implementing MFA for IAM users of AWS accounts with permissions to operate AWS resources handling communications with arrownet. This is because your arrownet communication settings can potentially affect other market participants. This prevents someone from inadvertently changing or deleting important AWS resources. • **Automate enforcement of access controls** • **Integrate with centralized federation provider** We recommend integrating the identity provider and the AWS Directory Service for more clear understanding of user access history. • **Enforce password requirements** We recommend implementing a strong password policy. • **Rotate credentials regularly** We recommend setting an expiration date for login passwords. This prevents unauthorized users from unnecessary access. • **Audit credentials periodically** We recommend performing periodic audits as you retrieve Credential Reports for IAM users from AWS Identity and Access Management.

Considerations	Questions	Guide
SEC2	**How do you control human access?**	• **Define human access requirements** As with [SEC1], we recommend that you follow the security rules defined by the market participant's organizations and establish access requirements for AWS IAM Roles based on their job functions. In addition to users involved in the construction and operations, make preparations of right IAM Roles in advance for users involved in responding to security incidents and those involved in audits. • **Grant least privileges** Grant users only the minimum privileges you have defined so as to reduce the risk of unauthorized access. • **Allocate unique credentials for each individual** • **Manage credentials based on user lifecycles** • **Automate credential management** • **Grant access through roles or federation** We recommend achieving this by integrating with a centralized federation provider (mentioned in [SEC1]).
SEC3	**How do you control programmatic access?**	• **Grant least privileges** • **Automate credential management** • **Grant access through roles or federation** • **Implement dynamic authentication** We recommend implementing the contents described in [SEC1—2], even if you are operating the AWS environment through programmatic access. • **Allocate unique credentials for each component** We recommend that you use IAM roles or federation instead of IAM users or static access keys to allow secure programmatic access.

Considerations	Questions	Guide
SEC4	**How do you detect and investigate security events?**	• **Define requirements for logs** Establish log storage and access control requirements that meet the security rules defined by the market participant's organization. • **Define requirements for metrics** We recommend using AWS CloudTrail to collect a history of AWS operations. We recommend assessing log metrics on Amazon CloudWatch Logs. An example of a log metric would be whether a particular log is being generated frequently within a short period of time. We also recommend enabling Amazon GuardDuty and any monitoring service that identifies unauthorized API calls and unexpected access to resources. • **Define requirements for alerts** • **Automate alerting on key indicators** Define which departments should receive alerts on security incidents and what the medium of communication should be. We recommend using Amazon SNS to send out alerts based on these definitions whenever CloudWatch Logs detect metrics anomalies or when Amazon GuardDuty identifies unauthorized activities. • **Configure service and application logging** Collect logs using Amazon CloudWatch Logs, VPC Flow Logs, and AWS CloudTrail. • **Analyze logs centrally** We recommend that you centralize, aggregate, and analyze these logs in S3. We also recommend that you use AWS Security Hub to improve visibility. • **Develop investigation processes** Refer to the content in [OPS5] and [OPS7].

Considerations	Questions	Guide
SEC5	**How do you defend against emerging security threats?**	• **Keep up-to-date with organizational, legal, and compliance requirements** • **Keep up to date with security threats** • **Define and prioritize risks using a threat model** Establish requirements that must be implemented in order to meet the security rules defined by the market participant's organization. • **Keep up-to-date with security best practices** • **Evaluate new security services and features regularly** • **Implement new security services and features** Refer to [PERF6], and consider using Amazon Inspector to detect unknown access.
SEC6	**How do you protect your networks?**	Assuming arrownet is a closed service, this reference architecture acts as a demilitarized zone (DMZ) for the market participant's workloads and protects its networks. You can also protect the market participant's workloads by using security groups or Network Access Control List (NACLs) to prevent unnecessary communications. Also, from the network protection perspective, refer to [SEC1] and [OPS3] and consider implementing measures to prevent human error (such as limiting manual operations using AWS Management Console).
SEC7	**How do you protect your computing resources?**	Assuming arrownet is a closed service, apply patches periodically to the EC2 OS to prevent unauthorized access from arrownet or from market participants. If it is also within the market participant's security rules, consider using AWS Systems Manager to automate the application of security patches.

Considerations	Questions	Guide
SEC8	**How do you classify your data?**	This document does not accumulate data on the market participant's dealings with JPX. Consider in advance the logs that will be accumulated. As these logs are generated from communication with JPX services, depending on the service type, they may contain confidential information from the market participant's point of view.
SEC9	**How do you protect your data at rest?**	We recommend encrypting logs for storage. Consider using the Customer-Managed Key feature of AWS Key Management Service to limit services and IAM accounts that have access to logs. This limits the possibility of accidentally changing, deleting, or moving log information. For even stronger protection, you can apply settings that prevent direct access to the storage services (such as S3) where the logs are stored.
SEC10	**How do you protect your data in transit?**	Follow the defined communication requirements of services provided by JPX.

Considerations	Questions	Guide
SEC 11	**How do you respond to security incidents?**	• **Identify key personnel and external resources** • **Identify tools** • **Develop incident response plans** • **Pre-deploy tools** For more details, refer to [OPS5]. • **Automate containment capability** Consider measures to contain the abnormalities on the market participants' end so that they do not flow to arrownet. These measures may include methods to stop the AWS Auto Scaling function and the EC2 instance that is used to set up a proxy server. • **Identify forensic capabilities** AWS provides a VPC Traffic Mirroring feature, which allows you to capture and mirror EC2 (used to set up a proxy server) instance network traffic. Please also consider both implementation method and location, taking into account regular traffic between market participants and arrownet, market participants' security rules, and cost efficiency. • **Pre-provision access** See the description in [SEC2]. • **Run game days** By conducting regular simulations, it is possible to uncover escalation paths that have not yet been updated (e.g., transfers within the organization) or reveal the absence of specific coping methods. Consider conducting simulations while referring to the description in [OPS5], [OPS7], [OPS8], [OPS9].

Considerations for Reliability

The following table describes a design guide based on Reliability considerations from the Well-Architected Framework.

Table 4 – Considerations for "Reliability"

Considerations	Questions	Guide
REL1	**How do you manage service limits?**	Resources that can be provisioned for each AWS account have quotas (previously called limits). Check for resources that can be provisioned with the Service Quotas. If you are seeing a lot of traffic with arrownet, there could be many large EC2 instances being used. In that case, pay close attention to the quotas. Also, check in advance to determine if you are able to cope with service limits and failovers that include AZ failures. If you cannot, increase the quota in advance.
REL2	**How do you manage your network topology?**	Refer to the scenario-specific architectures in this Implementation Guide for ways to increase availability when connecting arrownet to market participants' AWS environments and even on-premises environments.
REL3	**How does your system adapt to changes in demand?**	Originally arrownet has traffic bandwidth throttling, but following the contents in [OPS6], always monitor, accumulate, and analyze the metrics, keep optimizing number and size of EC2 instance according to adjusting in traffic. Auto Scaling function is very helpful as a way for instance replacement and to ensure base capacity across multiple Availability Zone. When starting or changing a connection, follow the JPX arrownet version2.0 guidelines and carry out a load test.

Considerations	Questions	Guide
REL4	**How do you monitor your resources?**	Monitor the communication status with arrownet using Amazon CloudWatch Metrics and Logs, according to the contents in [OPS6]. Use the AWS Service Health Dashboard and Personal Health Dashboard to monitor the AWS service itself for any issues.
REL5	**How do you implement change?**	Follow the contents in [OPS3], reflect the changes in the AWS CDK/AWS CloudFormation code and automate the deployment of these changes.

Considerations	Questions	Guide
REL6	**How do you back up data?**	• **Identify all data that needs to be backed up and perform backups or reproduce the data from sources** The scope of this guide involves the backing up of Amazon Elastic Block Store (EBS) (used by EC2) data using snapshots. • **Perform data backup automatically or reproduce the data from sources automatically** Use AWS Backup and create settings so that multiple generations of snapshots remain. If you already have a job management function, use the AWS Command Line Interface (CLI) from the job and implement a batch application that periodically manages the snapshot and its generation. • **Perform periodic recovery of the data to verify backup integrity and processes** This is related to the preparation of the playbook for [OPS5] and its utility check. We recommend that you verify beforehand that EC2 can be restored from a snapshot. • **Secure and encrypt backups or ensure the data is available from a secure source for reproduction** Snapshots are also encrypted as the EBS is encrypted.

Considerations	Questions	Guide
REL7	**How does your system withstand component failures?**	• **Monitor all layers of the workload to detect failures** As an external monitoring device, the health check function provided by each system/service accessible through arrownet is used to monitor whether the system/service is functioning normally from a market participant's point of view. • **Send notifications upon availability impacting events** According to the description in [OPS6] and [OPS7], the normality of the workload and operation is monitored, where a notification is sent if an error occurs. • **Implement loosely coupled dependencies** • **Implement graceful degradation to transform applicable hard dependencies into soft dependencies** • **Automate complete recovery because technology constraints exist in parts or all of the workload requiring a single location** • **Deploy the workload to multiple locations** • **Automate healing on all layers** If market participants use this connection as an important workload, have JPX provide two or more VIFs and prepare for any AWS Direct Connect route abnormalities. Also, as shown in this reference architecture, set up a proxy server that automatically recovers across multiple Availability Zones.
REL8	**How do you test resilience?**	Although guide is designed to provide resiliency, it is recommended that a sufficient amount of testing be performed in advance, as described in [OPS5] and [OPS8].

Considerations	Questions	Guide
REL9	**How do you plan for disaster recovery?**	At the time of writing, the answer to this question posed to the Well-Architected Framework is "Not applicable to workload" because arrownet does not support a wide-area disaster recovery beyond the Kanto Region.

Considerations for Performance Efficiency

The following table describes a design guide based on Performance Efficiency considerations from the Well-Architected Framework.

Table 5 – Considerations for "Performance Efficiency"

Considerations	Questions	Guide
PERF1	**How do you select the best performing architecture?**	We recommend that you design and construct according to this Implementation Guide. It is also necessary to follow the JPX guide for conducting load tests.
PERF2	**How do you select your compute solution?**	Select an EC2 instance size based on the purpose of the market participants while evaluating the communication performance. For more information about configuration options and retrieving of metrics, see Architecture Explanation. In addition, as described in [OPS6] and [OPS7], it is recommended to perform regular evaluations based on metrics and choose the optimal instance size.
PERF3	**How do you select your storage solution?**	Evaluate the EBS performance used by the EC2 instance to set up a proxy server, and select the EBS volume type based on the purpose of the market participants. Refer to [SEC8—9] for more information on the EBS encryption configuration option. We recommend that you perform regular evaluations based on metrics while considering the instance size to adjust for an optimal EBS volume type.

Considerations	Questions	Guide
PERF4	**How do you select your database solution?**	This Implementation Guide does not work with database solutions. Therefore, the answer to this question posed to the Well-Architected Framework is “Not applicable to workload”.
PERF5	**How do you configure your networking solution?**	• **Understand how networking impacts performance** • **Understand available product options** • **Evaluate available networking features** • **Leverage encryption offloading and load balancing** We recommend that you design and construct according to this Implementation Guide. • **Use minimal network ACLs** • **Choose network protocols to improve performance** We recommended that you use the security group attached to the proxy server and avoid the use of NACLs as much as possible. You may set the communication permission of the security group according to the communication requirements provided by each JPX service. • **Choose location based on network requirements** It is necessary to comply with the Financial Instruments and Exchange Act. Refer to the "Architecture Explanation" and select the appropriate Region. • **Optimize network configuration based on metrics** Refer to the description in [PERF2] to fine-tune the size of EC2 instances.

Considerations	Questions	Guide
PERF6	**How do you evolve your workload to take advantage of new releases?**	We embrace continuous innovation in our services. Check *What's New at AWS* and the *AWS Blog* regularly for more information. You can also consult with an AWS Solutions Architect. We recommend that you evaluate the impact of new feature releases on your existing workload, take an active approach, and implement a process that can evolve your workload performance.
PERF7	**How do you monitor your resources to ensure they are performing as expected?**	Refer to the content in [OPS3] to [OPS9].
PERF8	**How do you use trade-offs to improve performance?**	In this guide, the size of EC2 instances has been fine-tuned. If, however, you require storage, memory, or compute optimization, those instances may increase costs. For more details refer to [COST4].

Considerations for Cost Optimization

The following table describes a design guide based on Cost Optimization considerations from the Well-Architected Framework.

Table 6 – Considerations for "Cost Optimization"

Considerations	Questions	Guide
COST1	**How do you govern usage?**	• **Develop policies based on your organization requirements** • **Implement an account structure** • **Implement cost controls** • **Track project lifecycle** This Implementation Guide uses a dedicated AWS account to handle arrownet traffic. All costs incurred in this AWS account will be related to communication with arrownet, making cost management easier. • **Implement groups and roles** Refer to the content in [SEC1—3].
COST2	**How do you monitor usage and cost?**	As described in [COST1], all costs incurred in this AWS account are related to communication with arrownet. We recommend that you keep track of your costs with AWS Cost Explorer and plan your cost management with AWS Budgets. If the cost changes abruptly, it will be seen as a trigger that causes an abnormality in the environment. AWS Budgets will be triggered and SNS will send a notification to the administrator.
COST3	**How do you decommission resources?**	As described in [OPS3—8], we recommend decommission unnecessary resources according to the runbook/playbook procedures.

Considerations	Questions	Guide
COST4	**How do you evaluate cost when you select services?**	We recommend that you optimize the proxy server and conduct cost analysis as mentioned in [PERF2], [PERF3], and [PERF5].
COST5	**How do you meet cost targets when you select resource type and size?**	Refer to the content in [COST4].
COST6	**How do you use pricing models to reduce cost?**	• **Perform pricing model analysis** • **Implement different pricing models with low coverage** • **Implement pricing models for all components of this workload** We recommend reserving the capacity of instances and optimizing your costs. In order to achieve both of these at the same time, this guide recommends that you use Reserved Instances. • **Implement Regions based on cost** You must select the Tokyo Region in order to maintain compliance.
COST7	**How do you plan for data transfer charges?**	Refer to the content in [COST4].

Considerations	Questions	Guide
COST8	**How do you match supply of resources with demand?**	In this guide, we mention that an instance type is selected through PoC and design planning in advance, and as a result, resource demand is not expected to fluctuate rapidly in everyday use. Over the medium and the long term, however, you must evaluate the traffic metrics with arrownet. Depending on the results, take action such as changing the instance types to balance supply of resources with the actual traffic demands.
COST9	**How do you evaluate new services?**	Refer to the content in [PERF6].

Conclusion

Using AWS to connect with arrownet can help you increase business agility, optimize human resources, and maximize investment.

Business agility
Infrastructure and its resources can be allocated and changed at all times, making them more resistant to market fluctuations. This also reduces the time it takes for a user to bring a service to the market.

Human resource optimization
Using the AWS Cloud allows engineers to be relieved from miscellaneous tasks (such as working on data center applications)—allowing them to concentrate on differentiating tasks. In addition, there is no physical line application, and users are freed from interacting with related departments/company.

Investment maximization
By anticipating market fluctuations and eliminating IT investments that require several times the amount of IT resources normally required, users' investment plans are optimized. Users can use as many IT resources as they need with no upfront costs.

This Implementation Guide addresses the advance preparations and measures required for users to securely use a market infrastructure like arrownet. This guide allows users to establish secure connections with arrownet while taking advantage of the benefits of the AWS Cloud.

Contributors

Contributors to this document include:

- Yoshinobu Sawano, Solutions Architect, Amazon Web Services

Comments and Feedback

If you have any feedback about this guide, contact us using the email address below.

Email: aws-jp-refarchguide-jpxarrownet@amazon.com

Document Revisions

Date	Description
May 2020	First publication

Additional Resources

For additional information, see:

- ***AWS Well-Architected Framework***
  https://aws.amazon.com/jp/architecture/well-architected/
- ***Webinar Documents in Japanese***
  https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/
- **Japan Exchange Group, Inc (JPX)**
  https://www.jpx.co.jp/
- **arrownet**

  https://www.jpx.co.jp/systems/network/index.html

Notes

An understanding of AWS service components is required to use AWS, and to build an arrownet connection service. This appendix provides reference information on the services used in this guide.

1. Amazon Athena - https://aws.amazon.com/athena/
2. Amazon CloudWatch - https://aws.amazon.com/cloudwatch/
3. Amazon Elastic Compute Cloud (EC2) - https://aws.amazon.com/ec2/
4. Elastic Load Balancing - https://aws.amazon.com/elasticloadbalancing/
5. Amazon Elasticsearch Service - https://aws.amazon.com/elasticsearch-service/
6. Amazon GuardDuty - https://aws.amazon.com/guardduty/

7. Amazon Inspector - https://aws.amazon.com/inspector/
8. Amazon Kinesis – https://aws.amazon.com/kinesis/
9. Amazon QuickSight - https://aws.amazon.com/quicksight/
10. Amazon Simple Notification Service (SNS) - https://aws.amazon.com/sns/
11. Amazon Simple Storage Service (S3) - https://aws.amazon.com/s3/
12. Amazon S3 Glacier - https://aws.amazon.com/glacier/
13. Amazon Virtual Private Cloud (VPC) - https://aws.amazon.com/jp/vpc/
14. AWS Auto Scaling - https://aws.amazon.com/jp/autoscaling/
15. AWS Backup - https://aws.amazon.com/backup/
16. AWS Budgets - https://aws.amazon.com/aws-cost-management/aws-budgets/
17. Amazon AWS Cloud Development Kit (CDK) - https://docs.aws.amazon.com/cdk/latest/guide/what-is.html
18. AWS CloudTrail - https://aws.amazon.com/cloudtrail/
19. AWS CloudFormation - https://aws.amazon.com/cloudformation/
20. AWS CodeCommit https://aws.amazon.com/codecommit/
21. AWS Command Line Interface (CLI) - https://aws.amazon.com/cli/
22. AWS Config - https://aws.amazon.com/config/
23. AWS Cost Explorer - https://aws.amazon.com/aws-cost-management/aws-cost-explorer/
24. AWS Direct Connect - https://aws.amazon.com/jp/directconnect/
25. AWS Glue - https://aws.amazon.com/glue/
26. AWS Identity and Access Management (IAM) - https://aws.amazon.com/iam/
27. AWS Key Management Service (KMS) - https://aws.amazon.com/kms/
28. AWS Lambda - https://aws.amazon.com/lambda/
29. AWS Security Hub - https://aws.amazon.com/security-hub/
30. AWS Systems Manager - https://aws.amazon.com/systems-manager/
31. AWS Transit Gateway - https://aws.amazon.com/jp/transit-gateway/