Container Migration Methodology

November 2020

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

More and more customers are migrating their IT environments to the cloud. Their containerized application systems are migrated to the cloud with the overall project. Because container migration to the cloud has unique characteristics, customers need to design a targeted cloud migration solution. This whitepaper describes some common scenarios based on the experience of several source Kubernetes platforms migrating to the Amazon Elastic Kubernetes Service (Amazon EKS). The purpose of this whitepaper is to assist cloud architects and operations engineers with designing their own cloud migration[1] plan.

Introduction

With the increasing popularity of public cloud and cloud-native systems, containerization has become one of the fastest-developing IT fields that has received the most attention from the market in the past two years. At the beginning of container technology development, users mostly came from internet companies. Most traditional companies were still in the exploration and evaluation stage, or trying to deploy containers on a small scale with the help of open-source solutions. In the past two years, as the technical direction has become clearer, the container-related open-source community has been actively expanding containerization ecology and technology. The need for agile architecture and innovation in digital transformation among enterprises has driven container technologies to develop and implement rapidly.

The following chart from Grand View Research shows the growth of the application container market size in the United States.

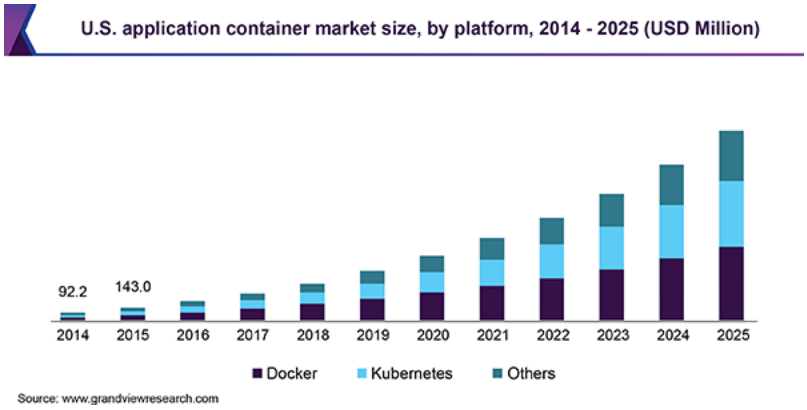

Figure 1 – U.S. application container market size

Container migration is a systematic project, which needs to consider the impact of business and technology on migration work. The container migration methodology covered in this whitepaper will guide architects and operations engineers to migrate containers from their source to an Amazon Web Services (AWS) container service.

In this whitepaper, we will discuss some migration situations, and how to complete the migration projects successfully. Subjects covered include assessment, initiation, and migration. We won't discuss detailed migration technologies in this article.

Why Migrate to the AWS Cloud?

There are five major reasons for companies to move to the AWS Cloud:

- **Agility and productivity improvement** — Customers who migrate their container platform to cloud-hosted services improve agility and productivity because containers have more advantages for agility than virtual machines, but maintain deep integration capabilities.
- **Cost savings** — Migrating to the cloud can initially save operating costs. Migrating to a container platform, for example; the AWS Fargate model, can further improve resource utilization and flexibility, which reduces costs even more.
- **Elasticity** — The customer can provision the container cluster that they need, knowing they can instantly scale up or down along with the needs of their business, which reduces cost and improves the customer's ability to meet their user's demands.
- **Faster innovation** — The container service on the cloud hosts the control plane, reduces the customer's operation and maintenance work, and can devote more resources to innovation.
- **Deploy globally in minutes** — Using AWS, customers can leverage 77 Availability Zones across 24 geographic Regions worldwide.

Why Migrate to Amazon Container Service?

There are many reasons for customers to migrate to Amazon container services, including the following:

- **DevOps driver** — The DevOps concept is widely accepted and applied to production, and the use of container technology can support the DevOps process to run faster. Using managed container technology to support DevOps on the cloud platform further improves efficiency. As a result, DevOps has accelerated the customer migration pace to containers.
- **Platform as a Service (PaaS) platform building** — The cloud platform provides many platform components for you to build your own applications. However, some customers need customized functions or components for application systems. They build their own customized PaaS platforms to make them flexible and efficient, and combine them with their own DevOps processes. Migration to the container platform not only improves efficiency, it can ensure that customers build on the same technology platform as their DevOps platform to reduce technical complexity.

- **Operations simplification** — The current container technology ecosystem is huge and wide ranging. The technical threshold for operations of the container orchestration engine is also very high, making it helpful for customers to reduce their workload and cost of operations by migrating to managed container services. Efficiency is further improved when integration is combined with other deeply integrated hosting services.
- **Expectation of the same user experience as native Kubernetes** — The widespread adoption of Kubernetes has led cloud providers to launch managed Kubernetes services to reduce customers' operations workloads. However, because the container ecosystem is in a state of rapid development, the open-source community is iterating very fast, and customers are eager to experience new features. Customers want the convenience of hosted services without losing the freedom brought by open source. The hosted Amazon EKS platform is adopted by many customers for its consistent user verification and open-source plug-ins with the upstream community.
- **Digital transformation** — Digital transformation is a high-level transformation based on digitization and digitalization to further elevate the company's core business and create a new business model. Digital transformation involves the development of digital technologies and support capabilities to create a vibrant digital business model. In this transformation, the upgrade and transformation of IT technology is important.
- **Internet of Things/Machine Learning (IoT/ML) innovation** — The continuous development of IoT and the rise of machine learning in various fields has led to a technological innovation. The two phases of an artificial intelligence (AI) project based on container environment, training model, and deployment are very promising:
    - **Model training** — Training AI models is a computationally intensive operation, and containers have proven to be beneficial in terms of scalable workloads and rapid communication with other nodes.
    - **Deployment** — Containers not only provide a portable, isolated, and consistent environment for the rapid deployment of ML and AI models, they also have the ability to change today's IT landscape by enabling enterprises to achieve their goals faster and better.
- **Deep integration with AWS** — AWS container services are deeply integrated with AWS by design. This enables your container applications to leverage the breadth and depth of AWS cloud technologies, including networking, security, and monitoring. AWS combines the agility of containers with the elasticity and security of the cloud.

- **Security and compliance** — AWS offers 210 security, compliance, and governance-related services and key features — about 40 more than the next largest cloud provider. AWS provides strong security isolation between your containers to ensure that you are running the latest security updates, and to give you the ability to set granular access permissions for every container.

AWS Capabilities for Containers

As an orchestration engine in the container field, Kubernetes already occupies a large portion of the market in the traditional field. Various vendors use Kubernetes-based hosting services to reduce the complexity of usage, operations, and maintenance. Kubernetes-related demands have become common in the process of customer migration to the cloud.

When running containers on AWS, you have two choices to make:

1. Whether or not you want to manage the server.
   - For serverless computing for containers, choose AWS Fargate.
   - If you need control over the installation, configuration, and management of your computing environment, choose Amazon Elastic Compute Cloud (Amazon EC2).
2. Which container orchestrator you want to use.
   - Amazon ECS is deeply integrated with the rest of the AWS platform capabilities and fully controlled by AWS, so we can ensure that every feature we launch works closely with ECS.
   - Amazon EKS is a managed Kubernetes service. EKS runs upstream Kubernetes and is certified Kubernetes conformant, so you can leverage all the benefits of open-source tooling from the community. You can migrate any standard Kubernetes application to EKS without needing to refactor your code.

The container images library is named Amazon Elastic Container Registry (Amazon ECR). AWS App Mesh provides you with application-level networking. It is the only service mesh that allows communication across multiple types of compute infrastructures such as Amazon EC2, Amazon ECS, AWS Fargate, and Amazon EKS.

AWS offers the broadest choice of container orchestrators, so you can run your containers on AWS regardless of your choice of tools or APIs.

Strategy for Container Migration

In this chapter, we will discuss how to formulate a container migration strategy. Before formulating a migration strategy, customers need to evaluate the migration project through a series of means and methods. Architects need to learn about the customer's technical details to help them choose specific migration methods to lay the foundation for the subsequent migration.

Assessment

Before formulating the container migration project strategy, evaluate the customer's migration preparation to ensure that the migration can solve the customer's problems, and to provide the basis for the customer to make decisions in key areas of the migration plan.

Business Capabilities

- **Business Target** — Evaluate business goals that the customer expects to achieve through container migration, such as accelerating the launch of business systems, cost savings, expected investment in the migration, and so on. Common roles involved in the evaluation include:
    - Business managers
    - Financial managers
    - Budget owners
    - Migration strategic decision makers
    - Stakeholders
- **People** — Understand the composition of the customer's IT personnel to determine who will be involved in the migration project, and to determine whether the customer should increase or adjust staffing.
    - **Technical scope** — Assess the skill set of the customer's IT team. Container migration may require support for multiple technologies, such as containers, Kubernetes, network, CI\CD tools, and more.
    - **Recruitment and training** — Assess whether the customer should recruit or train technical staff for the container migration project.
    - Common roles involved in staff assessment:
        - Human resources
        - Staffing specialists

        - People managers

- **Governance**
    - **Project management** — Evaluate which teams will participate in the container migration project, assign the person in charge of each team, and identify the final decision maker for the migration project decision chain. Evaluate the effectiveness of project management tools and communication mechanisms.
    - **Migration effect measurement** — Evaluate how the customer measures project results, whether they have specific indicators such as release frequency, cost reduction ratio, and so on.
    - Common roles involved in governance evaluation:
        - Chief Information Officers (CIOs)
        - Project managers
        - Enterprise architects

Technical Capabilities

- **Platform**
    - **Cloud platform** — Assess the customer's familiarity with the AWS Cloud platform, especially the key concepts and skills of container-related basic services. Managed container services have a dependency on the AWS infrastructure. The ability to use basic services determines the quality and speed of building the container platform, which directly affects the project's progress.
    - **Container platform** — Assess the customer's familiarity with the AWS container platform and their related skill set. There are many components and subsystems associated with the container platform, and there will be differences in concepts and usages, especially for managed container services.
    - **Assessment method**
        - **Immersion Day** — Immersion Day is a critical evaluation tool for you to:
            - Introduce the basic concepts and service design concepts of Amazon container services.
            - Introduce the ecology of Amazon container services.
            - Communicate and exchange container technologies with other parties.

- **Proof of Concept (PoC) —** Through PoC, you can understand the following:
  - The customer's familiarity with AWS container services such as:
    - How to create a cluster and node group
    - How to use ECR
    - Whether they fully understood Amazon EKS
    - Whether they understand the difference between Amazon EKS and a source Kubernetes cluster
  - **Pilot for technical verification**
    - EC2&Fargate model, managed or unmanaged node group
    - Container Network Interface (CNI), Application Load Balancing (ALB) ingress
  - **Test the demo applications for migration on AWS**
  - After PoC, the customer needs to know the following information:
    - The basics of Amazon EKS (cluster, role, node group, CNI, and so on)
    - The conceptual difference between Amazon EKS and other platforms
    - The best method to understand Amazon EKS
    - The ability to identify the applications that can be migrated immediately during the PoC process
    - Access your personnel effort for the migration project

- **Security and compliance** — Evaluate the customer's requirements on AWS infrastructure security, such as network and encryption measures.
  - **Evaluate the requirements of authority management** — Understand the customer's need for Identity Access Management (IAM), role-based access control (RBAC), and Pod implementation roles such as cluster IAM, node IAM, and Pod execution.
  - **Evaluate the security requirements related to service accounts** — Does the customer need a service account, and what are the minimum permissions? Does the certificate need to be isolated and reviewed?
- **Operations**

- **Monitoring** — The monitoring metrics to evaluate the customer's needs, the monitoring tools used, and the methods to build their own monitoring system.
- **Alarm** — Evaluate the customer's requirements for alarms, alarm indicators, and the impact of the alarm indicators' value range on the business.
- **Analysis** — Evaluate whether the customer has analysis requirements in the container operations field, such as attack analysis and error root cause analysis.
- **Release management** — Evaluate the tools and processes used by the customer for release management in the software life cycle, and determine the migration method or optimization plan for release through the evaluation.
- **Disaster tolerance** — Evaluate the customer's disaster tolerance requirements to determine what kind of disaster tolerance solution is suitable for them. Learn about the customer's current disaster recovery plan and give optimization suggestions.

Container Migration Maturity Model

The maturity model of container migration measures the difficulty of the customer's container migration project by evaluating from two dimensions: platform operations capability, and the technology stack of the source cluster. The strength of the operation and maintenance technology will affect the customer's use of the new container platform during and after the migration; the technology stack will affect the difficulty and workload of the migration. See Figure 2.

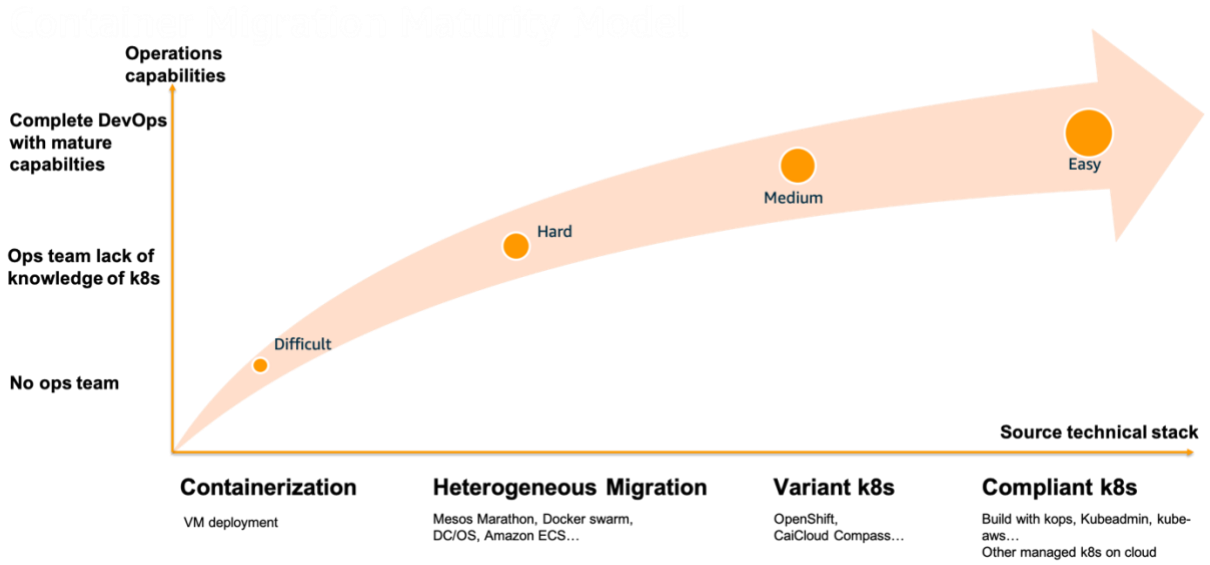

Figure 2 – Container Migration Maturity Model

The operation of containers and orchestration tools bring technical challenges:

- **Monitoring** — The content to monitor changes from the server to the container and service status, so monitoring methods and tools also change accordingly.
- **Logging** — Operations need to know the exact container and host from where application logs originate. The increases, decrease, and movement of containers also have impact on log collection.
- **Troubleshooting** — It is not possible to quickly judge and analyze container failures by adopting past behaviors. For example, in the container environment, the application cannot be directly modified and used, and the version needs to be re-released.
- **Security** — The rapid development of the community version creates an impact on security guarantee. Permission management poses new challenges to operations.
- **Network** — Network plug-ins are an important part of the container ecology, but they also increase the difficulty of network planning and design.

The functions of the operations platform and the capabilities of the operations team have a direct impact on the success of the container migration project. The degree of automation of the operations platform functions determines whether the customer can achieve the migration goal through automatic deployment during migration. The skill level of the operations team determines the automation degree of the operations platform. For example, if there is an automated process for CI\CD, the customer can send a release to the target cluster to complete the migration work (stateless service).

To evaluate the technology stack of the migration source cluster, we divide it into four scenarios:

- **Compatible Kubernetes** — One kind of compatible Kubernetes cluster is built on AWS through tools such as Kops or Kubeadmin. Another type is the Kubernetes platform hosted on other cloud providers. In terms of migration difficulty, a Kubernetes cluster built on AWS can be easily migrated to Amazon EKS, because the infrastructure is consistent and the related technologies are similar. Migrating from Kubernetes hosted by other cloud providers to Amazon EKS requires consideration on the migration of network plugins, Ingress, and image repository.
- **Variant of Kubernetes** — A typical representative of this type is OpenShift. Although it is also based on the Kubernetes engine, OpenShift itself provides Ingress. The CI/CD tool is provided by encapsulated Jenkins. The OpenShift "Deployment config" used when deploying the service is also different from the deployment of Kubernetes. A similar situation also exists in Kubernetes provided by some other third-party platforms.
- **Heterogeneous container orchestration engine** — Representatives of this type includes Apache Mesos and Docker Swarm. There are big differences in design concepts and implementation technologies. For example, Mesos uses two-stage scheduling. Kubernetes does not have this concept, which makes container migration projects across orchestration engines difficult.
- **Containerization** — This kind of migration is essentially the containerization of applications, which is the most primitive evolution from server deployment to containerized deployment. Such a project is technically the most difficult, because containerization is the first step in using the container platform, and there are many unexpected risks in the process. These risks include no developer support, lack of code deployment instruction documents, and the customer's micro-service requirements. These factors will seriously affect the progress of the migration.

Mobilization

In the mobilization phase, you will investigate the goals of the migration, build a migration team, assign roles and responsibilities, evaluate migration methods, and formulate a migration project plan.

Discover

Discovery is conducted mainly through questionnaires and interviews, to formulate a comprehensive understanding of the customer's business and technical goals and migration targets.

Discover Business Information (DBI)

- What is the target application system for migration? Which business unit does it belong to? Do they have any important business activities recently?
- **Migration cycle** — When does it start? What is the time span? Is there a clear deadline?
- **Migration expectations** — What is the goal to achieve? Does the customer have any clear metrics?
- **Personnel** — What is the number of personnel responsible for the migration? What kind of skills do they possess? Which modules are they responsible for?
- **Cost** — What is the labor cost? Migration cost? Dual environment parallel run cost? Target cluster planning cost?

Discover Technical Information (DTI)

- Where is the platform of the source cluster?
- What is the source cluster's actual usage of computing, storage, and network resources?
- Is the application stateful or stateless?
- What are the dependencies among applications?
- What is the technology stack used by the platform where the source cluster is located?
- Is there any source container cluster-specific information? (See the Appendix in this document.)

Report

A research report provides important information for selecting migration methods and output solutions. The report content should include but is not limited to the following:

- Cluster information
- Images repository
- Log collection subsystem information
- Monitoring subsystem information
- CI/CD subsystem information
- Business impact

Choose a Migration Method

Discuss the migration method with the customer based on the research and evaluation information.

Based on the analysis of the container migration maturity model, you can recommend a suitable migration method to customer:

- **Compatible Kubernetes** — If the source cluster is built on AWS, the application can be migrated through tools, regardless of whether it is stateful or stateless.

  If the source cluster is built on the platform of other cloud providers, stateless applications can be migrated through tools, while stateful applications must be migrated with the help of third-party partner software.

- **Variant Kubernetes** — This migration scenario has a certain degree of complexity and must be analyzed based on the specific source platform.

- **Heterogeneous container orchestration engine** — If the source container cluster currently uses a non-Kubernetes orchestration engine such as Mesos, the complexity of the migration project will be higher, because Mesos uses two-stage scheduling. To run and schedule the service, you must install the Marathon framework. From the design concept to the specific deployment, different container orchestration engines differ a lot, so this type of migration project becomes very complicated.

- **Containerization** — Containerization may be the most complicated scenario of all, because application containerization is affected by many factors. You must investigate and sort out how applications are containerized, and redesign the containerized architecture. Such projects require a lot of human resource investment.

- **With a CI/CD system** — The release process targeting the Amazon EKS cluster can be started directly through the CI/CD system. The prerequisite is to complete the preparation of the necessary infrastructure on AWS, such as configuring and labeling the network and working nodes.

Planning

In the planning stage, the main goal is to formulate a guidance plan for the migration project.

Migration Plan

In the overall plan of the project, you must reflect the scope, cycle, resource planning, current problems and risks, participating teams and personnel, project management and

coordination, and communication mechanisms. We recommend that you use project management best practices and agile delivery.

Include the following in the project plan:

- Review project management methods, tools, and capabilities for gap analysis.
- Define project management methods and tools, and how to use them in the project.
- Define project communication methods and problem escalation mechanisms.
- Develop a project task scheduling table, and clarify project risks and solutions.
- Decide the composition of the migration team, and clarify the responsibilities of the team.
- Outline the resources and costs required to migrate the target environment to AWS.

Technical Plan

In the technical planning phase, you must plan the excess cost, project schedule, and scope of the migration. This includes determining the application migration sequence, obtaining relevant data of the application to migrate, and analyzing the dependencies of the applications. Include the following:

- Discover the application dependency, which is critical for project prioritization and planning.
- Clarify the migration priority of the applications, and select the appropriate application system for migration verification.

AWS Landing Zone

In general migration projects, when the customer uses Landing Zone, AWS provides predefined configuration for initial structure for the AWS account, network, identity permissions, and billing framework. For container projects, you must plan for the IP segment of the Pod.

- **Account structure** — Define an initial multi-account structure and pre-configured baseline security, which can be easily introduced into your organizational model.

- **Network structure** — Provide basic network configurations to support the most common network isolation mode, achieve basic network connection between AWS and the local network, and provide user-configurable network access and management options. Plan for the Amazon EKS cluster network and the Pod IP pool based on the characteristics of the CNI network plug-in.
- **Account security baseline** — By default, the initial security baseline includes the following settings:
  - AWS CloudTrail
  - AWS Config
  - AWS IAM
  - Cross account access
  - Amazon VPC
  - Amazon GuardDuty
- **AWS user access management —** Provide a framework for cross-account user identity and access management (based on Microsoft Active Directory), centralized cost management, and reporting. Design the creation and management of users and permissions for the Amazon EKS cluster.

Skills/Cloud Center of Excellence (CCoE)

When preparing for container migration, it is crucial to train a certain number of people who already have AWS and Amazon EKS experience. During the migration process, the CCoE team will lead your company's technical team to conduct the migration work.

- **Goal** — Establish a migration team with experience in AWS and Amazon EKS. The migration team has the ability to implement container migration solutions.
- **Output** — Design how the CCoE team will lead and perform the migration task. Export AWS and Amazon EKS training plans to improve team skills.

Operations Model

To design the operations model for the container cluster environment to be migrated, start with the following:

- **AWS basic environment management** — The container environment is built on the AWS infrastructure. The operations of the basic environment are a necessary skill set. Customers must operate and maintain computing, storage, network, and permissions with managed services to reduce the workload.

- **Container cluster operations** — Worker node management (managed and unmanaged), worker node upgrade methods, dynamic scaling of work nodes, Pod capacity management, application deployment, and so on.
- **Monitoring** — Monitor the basic environment and Pod resource usage data, the status of hosts and Pod, and the status of application services.
- **Logging** — Select a log processing solution; decide how to collect host system logs and Pod application logs; determine how to process those logs.
- **Release management (DevOps)**:
    - **Version management process** — The basic process and functions of new version release.
    - **DevOps** — Tools for continuous integration and continuous deployment; code warehouse.
- **Change management** — This phase includes change management tool deployment and process description. The migration of the container project may lead to changes in the original process, and it needs to be explained here.

Security and Compliance

Different customers have different requirements for safety. The security plan is developed for the specific requirements of the customer. The best practice for security design is as follows:

Cluster Design

- For cloud infrastructure security, see Secure Cloud Computing Architecture (SCCA) on AWS GovCloud (US)
- IAM Roles — User roles, resource roles, and Pod roles
- Managed or unmanaged Node Groups
- Control SSH login
- EC2 security group — Security group reference and port opening between services

Network

- Network isolation — VPC, subnet, AWS PrivateLink, VPC peering
- Restrict network access to API server endpoint
- Open the private endpoint of the API server

- Protection service load balancers — Network Load Balancer (NLB) or Application Load Balancer (ALB)? ALB ingress or Nginx ingress?

Images

- Build secure images — Content addressable image identifier (CAIID)
- Use vulnerability scanning — Images scanner tools
- Image Repository — Use private image repository

Runtime Security

A Pod with excessive permissions that is infiltrated poses an extreme threat.

- **Namespace** — Provide scoping for cluster objects; allow fine-grained cluster object management.
- **RBAC** — Provide the standard method to manage authorization for the Kubernetes API endpoints. Creating and managing comprehensive RBAC roles following the principle of least privilege, in addition to performing regular audits on how those roles are delegated with role bindings, provides some most critical protections possible for your EKS clusters. This practice protects customers from both external attackers and internal misconfigurations or accidents.
- **Restrict the runtime permissions of the container** — By following the principle of least privilege and minimizing the capabilities of your cluster's running containers, you can greatly reduce the level of damage caused by malicious containers and misbehaving applications.
- **Pod security strategy** — Provide a method to enforce best practices, including not running as the root user, not sharing the host node's process or network space, no access to the host filesystem, enforcing SELinux, and other options, to minimize container runtime privileges.

Migration

In the mobilization phase, you planned the operation, security, and platform functions to achieve scale operations. In the migration phase, you can implement the migration plan with simple application migration, as requested by the customer, to accumulate migration experience and help subsequent migration work.

Design

Formulate the target AWS architecture, application architecture, operation, and maintenance process documents in the design phase of migration, based on previous research and planning in the mobilization phase.

- **Migration plan** — Determine AWS architecture, application architecture, operations process, and cutover plan based on Well Architectured and Amazon EKS Best Practice.
- **Test plan** — Output a functional test and integration test plan.
- **Cutover plan** — Define a process plan for cutover transaction traffic, describing the cutover process in detail.
- **Rollback plan** — If the migration cutover is unsuccessful, it can be rolled back to reduce the impact.

Migrate

In the migration verification process of previous projects, the customer migrated following the planned method, and the CCoE team gained valuable experience. The CCoE team can lead other implementation teams to perform the specific operations and steps of the migration. You can use automated processes or tools to speed up the migration.

- **Build the environment** — Build the basic environment and related operations subsystems following the migration plan.
- **Use automated processes or tools for migration** — Migration can be done with the help of CI\CD tools or other tools for migrating container applications, such as Velero.
- **Checklist** — Follow the checklist to confirm whether the migration has been completed before cutover. The checklist may vary for the customer's specific scenarios.

Validation

In the validation phase, each application must go through a series of specific tests (for example, construction verification, function, performance, and disaster recovery).

- Functional verification
- Performance verification
- Disaster recovery

After all test items have passed, you can enter the Cutover phase.

Cutover

Deploy the cutover plan, switch the transaction flow to the new system, and observe closely. If you find any abnormalities, run the rollback plan.

Items to consider:

- Output the playbook and runbook before cutover
- Exercise cutover process
- The CCoE plays a key role in sharing best practices and lessons learned across different migration teams

Glossary

- **Cloud Center of Excellence (CCoE)** — A diverse team of key members who play the primary role in establishing the migration timeline, and evangelize about moving to the cloud.
- **Landing Zone** — The initial destination area established on AWS to host the first applications, and ensure they have been migrated successfully.
- **Mobilization** — The stage in the migration process in which roles and responsibilities are assigned, an in-depth portfolio assessment is conducted, and a small number of selected applications are migrated to the cloud.
- **Operation** — The stage in the migration process when most of the portfolio has been migrated to the cloud and is optimized for peak performance.

Contributors

- Wang Jianli, Migration SA, Amazon Web Services
- Walkley He, Container Specialist SA, Amazon Web Services

Appendix

Container cluster-specific information. This information is for the survey form of the user source cluster before migration.

Cluster information

- Docker version
- Orchestration engine? K8S\Mesos
- Orchestration engine version?
- Is the K8s version compatible with the application?

- Is there any custom code in the compilation engine?
- What is the cluster size?
- Where is cluster deployed? IDC\Cloud
- What platform is deployed on? Rancher\OpenShift\Kops\ACK\GKE\AKS\Kubeadm
- Is there a Windows node?
- What plug-in is used? CNI, CSI, Cluster Autoscaler, Ingress, etc.
- Is it a stateful application?
- What type of storage driver is used?
- How much data is stored in the local disk?
- What is the full path of the data storage?
- Which type of storage is used? Volume, CSI, EFS, etc.
- Which type of network is used? Flannel/Calico/McVlan
- How is service discovery achieved? ZK, Eureka
- Which framework does the microservice use? Dubbo/ Spring cloud

Log collection solution

- Which log collection tools are used? Filebeat, Flume, Fluentd, etc.
- What is the deployment method? Build in Image/ Sidecar

Monitoring solution

- Which monitoring tool is used? Prometheus/Zabbix
- Which kind of dashboard is used? Kibana/Grafana
- What are metrics needed to monitor?

CI/CD solution

- Which pipeline solution is used? Jenkins/Spinnaker/GitOps
- Which package tool is used? Helm chart / Kustomize
- What are the source code repositories? GitHub/Gitlab/SVN

Image Repository

- Which image repository is used? Docker Repository/Harbor

- How many images are there?
- Business impact
- How long is the business interruption?
- Is there any dependencies between business systems?
- Is it feasible to migrate in stages?

Document Revisions

Date	Description
November 2020	First publication

Notes

[1] Migrate with AWS

[2] *An Overview of the AWS Cloud Adoption Framework* (whitepaper)

[3] AWS Simple Monthly Calculator

[4] AWS Pricing Calculator

[5] AWS Security Whitepapers & Guides