CloudHedge

IBM-AIX Workload Modernization to Linux for Enterprises using OmniDeq(TM)

Red Hat Certified Technology

redhat. TECHNOLOGY PARTNER

aws PARTNER Migration & Modernization Software

About CloudHedge

CloudHedge's OmniDeq(TM) - Automated App Modernization Platform powered by R6Ai(TM) is the world's only tool that modernizes legacy applications into containers by reHost/ reFactor/rePlatform and creates a defined path for organizations to leverage Container services and adopt microservices.

R6Ai™

Introduction to AIX

AIX (Advanced Interactive eXecutive) is IBM's homegrown UNIX operating system. First introduced by IBM in 1986 for the IBM RT PC RISC workstation, AIX now supports or has supported a wide variety of hardware platforms, including the IBM RS/6000 series and later POWER and PowerPC-based systems, IBM System I, System/370 mainframes.

In its existence of over 3 decades, AIX coupled with IBM Power systems enjoyed huge popularity due to its security, performance, and reliability. Even as Linux on x86 architecture started becoming more popular, today thousands of global enterprises still run their mission-critical enterprise workloads on AIX servers within their datacenters.

The Need to Modernize

When seen in isolation just like any time-proven technology the AIX servers continue to work deliver on their original promise of performance and reliability, to support heavy enterprise workloads. However, it cannot be seen in isolation considering the global trend of enterprises to get out of the data center model (convert CapEx heavy to OpEx). AIX workloads in this context become the roadblock: you cannot really move to cloud without moving the mission-critical apps running on AIX and yet there seems to be no easy way to just move them as-is.

Another major catalyst for modernization is the use of containers and managed container orchestration services on the cloud using Kubernetes. Linux is the most prominent base for containers. This means that AIX workloads that could benefit from cloud and container technology need to be first ported to Linux.

Cost of maintenance, availability of skilled administrators to manage AIX servers, and unsupported modern enterprise software's are some of the other reasons why an enterprise would look for migration/modernization of their AIX workloads.

The Challenges

The biggest challenge to move workloads out of AIX is rooted in what is called Endianness in computing parlance. While the UNIX based systems like AIX follow the "big endian" byte ordering, x86 based environments running Linux operating systems use "little endian" byte ordering.

This grassroot difference in how the values are stored in each architecture poses significant challenges in lift & shift solutions for migration of AIX workloads. This is the primary reason why the majority of cloud migration tools (including those provided by some of the largest cloud service providers) limit support to Linux and in some cases Windows workloads.

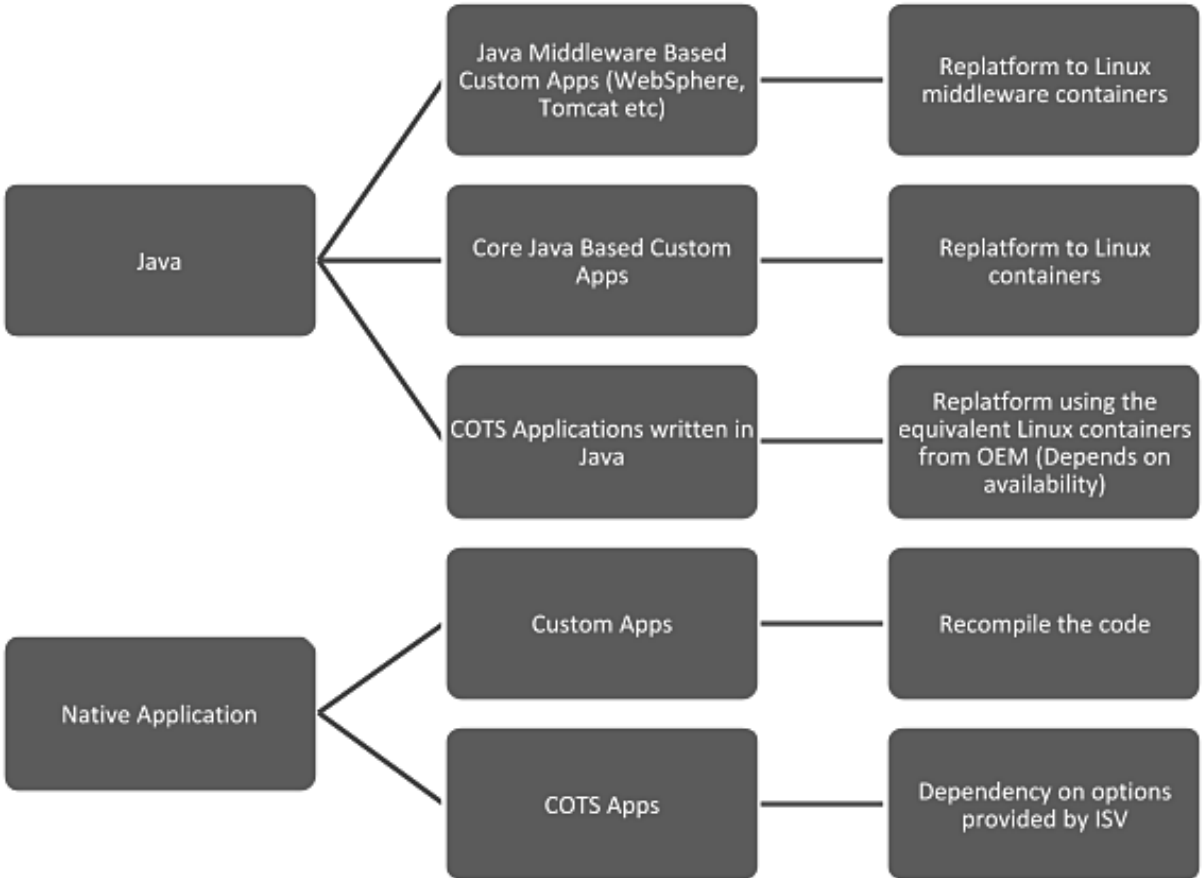
The Solution - Divide and Conquer

While the differences in the underlying instruction set between AIX and Linux make direct porting difficult, let us try to break the problem down.

The applications running on AIX can be categorized as follows:

- Java Applications
  - Java Middleware Based Custom Apps (WebSphere, Tomcat, etc.)
  - Core Java-Based Custom Apps
  - COTS Applications written in Java

- Native Application
  - Custom Apps
  - COTS Apps



Java

Java programming language (Another technology that assumed prominence is the 90's and 2000) has the advantage of being an interpreted language. This means that the applications built using Java get compiled into a bytecode that runs on runtime and can essentially be ported to any other environment with compatible runtime. This makes porting java applications a possibility.

Another level of abstraction comes in the form of the middleware layer. WebSphere Application Server is one of the favorites when it comes to running the enterprise Java & J2EE workloads on AIX for some of the largest fortune 500 companies across the globe. The ability to transform a WebSphere application with all of its dependencies and configuration into a WebSphere container running on Linux can address the majority of this legacy estate of applications.

A similar approach will work for core Java applications in re-platforming to Linux. However, in some cases, the application itself might rely on native services for its overall functionality and which may require remediation.

COTS Java Applications (typically software from ISV's) pose a business challenge, more than a technical one. As far as the technical solution goes, the COTS Java applications can be handled the same way as the Custom Java Applications. However, ensuring that the ISV will continue to support such environments is uncertain. ISV's may have the following stance:

- ISV may refuse support for containerized versions of their product
- ISV may plan to release container support in future
- ISV may have released a containerized version of the software and insist on supporting only those official container images

CloudHedge Partners:

- AWS
- Microsoft Azure
- Google Cloud
- VMware Tanzu
- Red Hat OpenShift
- IBM Cloud Pak
- TCS CUBO

Native Applications

Native applications are compiled for the particular hardware and instruction set on which they run. These typically are hard to solve as the only way to make these applications work in a different environment is to recompile from the source code. This in turn depends on having access to source code as well as the quality of documentation and know-how available to recompile it for a different platform. The availability of equivalent libraries and dependencies on the chosen target platform is another major challenge.

Native COTS applications are probably the hardest as the source code would not be readily available and hence the ISV has to be relied on to provide a solution.

Case in Point

While we break down the AIX landscape it is also important to consider the 80:20 rule. Given the prevalence of AIX+WebSphere combination in the market, a solution that addresses AIX+WebSphere to Linux+WebSphere can cover a significant portion of the business-critical workloads.

WebSphere workloads themselves can be classified into the following categories:

StandAlone WebSphere to tWAS Container

- These are typically utilized in development or QA environments.
- A container variant of WebSphere Traditional is published by IBM and is an ideal target for these environments.

WebSphere Network Deployment to tWAS Container

- These are clustered WebSphere deployments, which are most prevalent in customer environments.
- The deployment manager does most of the clustering and replication functions for various WebSphere nodes (worker nodes).
- While a container variant of WebSphere Traditional exists, it would typically represent one of the worker nodes in the cluster.
- The overall clustering, load balancing, routing functions need to be configured at the Kubernetes level, thereby removing the need for the deployment manager.

WebSphere ND to updated version of tWAS

- In this case, the WebSphere application server may need a version upgrade for compliance or support reasons.
- This would involve application code level refactoring since certain features may get deprecated across JDK version changes and need manual remediation.

WebSphere ND to WebSphere Liberty Container

- While the WebSphere traditional containers might serve the purpose, WebSphere Liberty profile containers are purpose-built for the cloud.
- These light-weight WebSphere containers are the ideal choice for running WebSphere workloads in containers and managed using Kubernetes.
- This transformation would involve discovering and analyzing WebSphere applications along with all of their configuration (Security, JDBC providers, Messaging queues, etc.), to be converted into a WebSphere Liberty container.

For AIX workloads #4 above is the best solution as it allows for migration and modernization of the applications in a single step, without engaging in expensive and time consuming refactoring at the code level.

CloudHedge Solution Approach

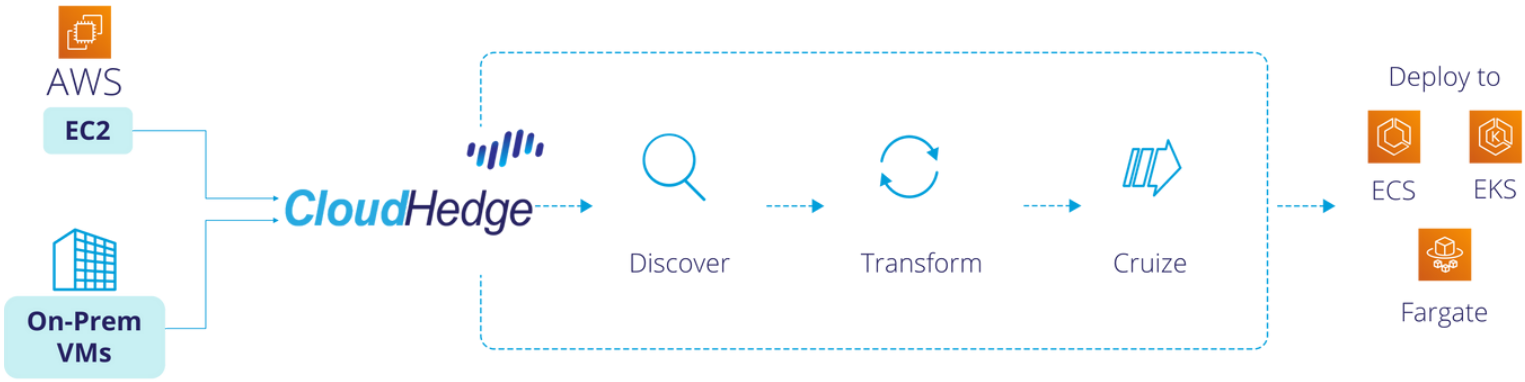
CloudHedge provides a 3-step solution that automates the end-to-end transformation of workloads using OmniDeq(TM):

- **Discover:** A deep discovery and analysis of the application landscape to providing insights on complexity and aid in decision making.
- **Transform:** The application and its configuration gathered during discovery is converted into the container using automation (re-platforming from WebSphere+AIX to WebSphere+Linux in the process).
- **Cruize:** The containers built from legacy applications can be deployed using automation (single click deploy) to any target Kubernetes service without the need for writing deployment configuration manually.

To address security compliance in highly secured environments, we propose the use of CloudHedge OmniDeq(TM) Enterprise edition which is installed within the customers' datacenter.

Once installed, the product does not require access to external internet and can work in a completely air-gapped environment.

Modernization of AIX Workloads using OmniDeq(TM)



Conclusion

CloudHedge with its automated containerization capability can allow customers to take their AIX workloads and replatform them to run as Linux-based containers. This further opens up the possibilities of running and operating these workloads in a variety of Cloud or Kubernetes environments, thereby leveraging the benefits of cloud and containers while reducing the cost of operations and maintenance.

CloudHedge transforms clients' business, operating, and technology models to be cloud-ready through its innovative OmniDeq™ platform powered by R6Ai™ consisting of – Discover™, Transform™, and Cruize™.
CloudHedge assists clients to:

- Envision, build and run efficient businesses in cloud,
- Modernizes monolithic applications to cloud-native by leveraging automated re-factoring and containerization technology

Headquartered in India, with a global presence in Singapore, Netherlands, and US, CloudHedge has deep partnerships with Amazon Web Services (AWS), VMware, Microsoft, Google Cloud Platform, Red Hat, Oracle, and IBM Cloud.