Deploying Liferay Digital Experience Platform in OpenShift on IBM Cloud

Liferay

Table of Contents

Introduction 1
Reference Architecture 1
Overview 1
Sizing 4
Implementation Details 5
Firewall 5
Load Balancer 5
Web Tier 6
Application Tier 6
Database Tier 7
Search Tier 7
Liferay DXP Specific Considerations 8
File and Database Storage 8
Search 8
Cloud Architecture Considerations 9
Security 9
SSL/TLS 9
Data Encryption 10
Autoscaling 10
High Availability and Disaster Recovery 10
Automated Setup 11
Backup Schedule and Replication 11
Data Recovery 12
Putting It All Together 12
Networking (Liferay clustering) 12
Cloud vs Bare Metal Performance 13
Summary 13
Disclaimer 13
Moving Forward 14
Liferay DXP Cloud 14
Liferay Global Services 14

Introduction

Liferay Digital Experience Platform (DXP) can be deployed into a variety of infrastructures and cloud-based environments. The Liferay Engineering and Global Services teams have extensive experience deploying and managing infrastructure in cloud environments, including OpenShift on IBM Cloud. Their accumulated experience and best practices are described in this comprehensive reference guide for Liferay Digital Experience Platform (DXP).

This document provides an initial environment for IBM Cloud OpenShift deployments and can be altered depending on your specific requirements for fault tolerance, scalability, and other infrastructure and non-functional requirements. The reference architecture is based on the base reference architecture as described in the Deployment Checklist with IBM Cloud OpenShift-specific technologies applied.

For additional hands on support, the Liferay Global Services team also has a specialized Go Live package that can help with your pre-production tuning and configuration.

Reference Architecture

The selection of an appropriate architecture is one of the first decisions in your deployment path. To select an appropriate architecture, you must consider:

- Information Security: securing your hardware and sensitive information against malicious attacks and intrusions
- Performance: supporting your desired number of total users, concurrent transactions, etc. to a level that matches your requirements
- Fault Tolerance: maintaining uptime during unexpected failures or scheduled maintenance periods
- Flexibility and Scalability: designing an expandable architecture to support additional features and users without significant redesign

Overview

The reference architecture depicted in Figure 1 appears complex but it provides high levels of fault tolerance and flexibility.

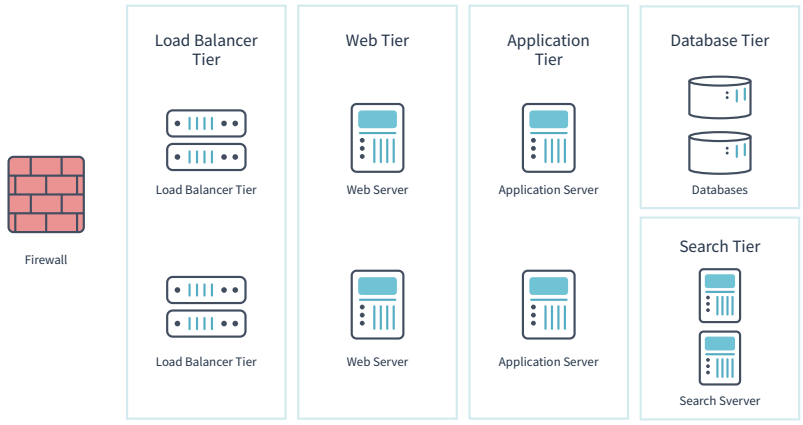

Figure 1: Liferay DXP Reference Architecture

The architecture contains the following tiers:

- Firewall
  - Provides intrusion detection and prevention
- Load Balancer Tier
  - Ensures smooth distribution of load between multiple web server resources and the underlying application servers
- Web Server Tier
  - Delivers static content elements like images, rich media, CSS files, etc.
  - Provides integration modules to single sign on solutions like CA Netegrity, Oracle Identity, etc.
  - Handles custom routing needs
- Application Tier
  - Hosts Liferay DXP on supported application servers like Tomcat, JBoss, GlassFish, Oracle AS, Oracle/BEA Weblogic and IBM Websphere*
- Database Tier:
  - Hosts Liferay supported database servers like MariaDB, Oracle, MS SQL Server, IBM Db2 and PostgreSQL*
- Search Tier:
  - Hosts Liferay supported search servers like Elasticsearch or Solr*

*Please see Liferay DXP Compatibility Matrix for a complete list of supported application servers, databases, search engines, etc.

Note that IBM Cloud's OpenShift service provides a layer of abstraction on top of Kubernetes. OpenShift lets you take advantage of application containerization and orchestration without having to worry about all of the details involved with direct Kubernetes management. The Liferay DXP reference architecture presented in this document includes several containerized applications and deployments. These include Liferay DXP, Apache HTTP server, and Elasticsearch.

Here's a diagram of the Kubernetes resources of the Liferay OpenShift reference architecture:

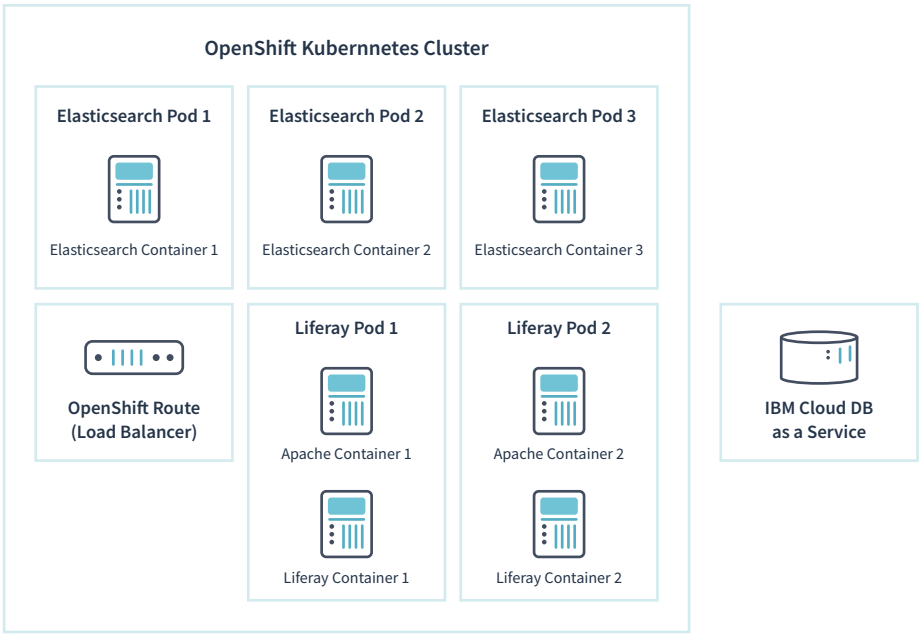

Figure 2: Liferay DXP OpenShift reference architecture.

The pods shown in this diagram are assigned to available OpenShift worker nodes (VMs) by the Kubernetes system. By default, three worker nodes are available but more can be added if needed.

While it's possible to run a database server such as MariaDB or PostgreSQL in your OpenShift cluster, we recommend against this. Instead, we recommend using a database as a service provider such as IBM Cloud's PostgreSQL as a service or Db2 as a service offering. Using a database as a service offering provides you with a simpler database management experience and lets you take advantage of automated database backups and optimized security configurations.

Sizing

When setting up an OpenShift cluster on IBM Cloud, you must select at least one of the six worldwide zones to use for your worker nodes. Ideally, your worker nodes should be created in several availability zones or locations to increase the level of resiliency against data center outages.

You can choose from a variety of hardware specifications for an OpenShift worker node. Here's one tested OpenShift worker node hardware specification:

- 8 vCPUs
- 32GB RAM
- b3c.8x32 sized VM
- RHEL 7
- 25GB SSD primary disk
- 100GB SSD secondary disk
- 1Gbps network speed

See cloud.ibm.com/docs/containers?topic=containers-planning_worker_nodes for more information on IBM Cloud's VM types, such as b3c.8x32.

For each layer of the logical architecture, we leveraged either IBM Cloud Services or pods running appropriate containers:

- Firewall
  - IBM Cloud Security Groups
- Load Balancer Tier
  - IBM Cloud OpenShift Routes (with an underlying HAProxy implementation)
- Web Tier
  - Primarily forwards traffic to the underlying application servers. However, it also provides caching, compression, and other capabilities.
  - 2 containerized Apache web servers. Each Apache web server will share a pod with the application server running Liferay DXP in the OpenShift Kubernetes cluster.
  - IBM Cloud CDN can be used for some functions at this tier. See www.ibm.com/cloud/cdn for details.

- Application Tier
  - Represents the workhorse of the architecture.
  - 2 containerized Liferay Tomcat bundles. Each Tomcat application server running Liferay DXP in the OpenShift Kubernetes cluster will share a pod with an Apache web server in the OpenShift Kubernetes cluster.
    - Note: Each Liferay container was configured to use up to a maximum of 8 vCPUs. You should configure the container to consume the appropriate number of vCPUs as allowed in your Liferay Enterprise Subscription
- Database Tier
  - IBM Db2 on Cloud or IBM Postgres on Cloud
- Search Tier
  - 3 containerized Elasticsearch servers running in the OpenShift Kubernetes cluster

Note that the number of each resource (web server, application server, database server, search server) depends on your requirements.

You must also select the number of worker nodes to create in each zone of the worker pool. 2 is the minimum, 3 is a good default. We're using the Liferay DXP Docker images to bootstrap deployments and ensure consistency.

Implementation Details

Firewall

IBM Cloud security groups are sets of IP filter rules that define how to handle incoming (ingress) and outgoing (egress) traffic to both the public and private interfaces of a virtual server instance. These IP filter rules may be used to isolate your entire infrastructure from the public internet and isolate layers such as the web tier and application tier according to your specific security requirements. See Getting started with IBM Security Groups and About IBM Security Groups for further details.

Load Balancer

IBM Cloud OpenShift routes allow you to manage external access to HTTP services in a Kubernetes cluster. OpenShift routes are implemented by HAProxy. They allow traffic to be distributed in an even manner between servers. When configuring a load balancer, you can define the protocols and ports your application is listening on, choose a load balancing method, enable sticky sessions, and more.

Different front-end (for incoming traffic to the load balancer) and back-end (for outgoing traffic from the load balancer) protocols and ports can be configured. IBM Cloud load balancers terminate incoming HTTPS connections so they can communicate in plain-text HTTP with back-end applications servers if HTTP is selected as the back-end protocol. See OpenShift Routes and Securing OpenShift Routes for further details.

Web Tier

The IBM Cloud web tier consists of two components:

1. (Optional) The IBM Cloud Content Delivery network (CDN): Content Delivery Network - Overview
2. The web tier also includes IBM Cloud VMs with Apache web server Docker containers running in the Kubernetes cluster. The IBM Cloud CDN is responsible for delivering static content, while the Apache web server Docker containers are deployed in the same pods as the application servers (the Liferay DXP Tomcat application server Docker containers). Incoming requests to the Apache web servers are forwarded to the corresponding Liferay Tomcat application servers.

In this reference architecture, since each Apache web server shares a pod with a Liferay Tomcat bundle, the Apache web server and Liferay Tomcat bundles run on the same IBM Cloud OpenShift worker node. Note that in other Liferay system architectures, the servers in the web tier are sometimes placed in separate VMs from those in the application tier. That approach makes the most sense when Apache is serving as the load balancer for the Liferay system. However, this is not the case here. Apache will mostly be used for serving static content so it's not necessary for Apache and Liferay to run in separate pods.

The Apache web servers can be configured to perform desired work such as compression, caching, etc. and then to forward requests to the appropriate application server in the next tier. Liferay DXP should be configured accordingly by disabling any unnecessary servlet filters in Liferay (e.g., the gzip compression filter). The Apache web servers can also be configured to server static resources like images and CSS files.

Application Tier

This tier consists of IBM Cloud OpenShift worker nodes running application server Docker containers on which Liferay DXP Tomcat bundles have been deployed. When working with OpenShift on IBM Cloud, your applications are deployed

in Docker containers running on Red Hat Enterprise Linux (RHEL) VMs, orchestrated by Kubernetes. As mentioned above, each Apache web server Docker container is deployed with a Liferay Tomcat bundle in a single pod. In this respect, the web tier and the application tier of this reference architecture overlap.

If you need to scale your Liferay DXP system to support more concurrent users, remember that the application tier is the tier that consumes the most system resources. To handle a higher user load, you should increase the number of replicas of your DXP pods as well as add one or more OpenShift worker nodes.

For testing the reference architecture, we specified a max of 8 (virtual) CPUs per Liferay node.

Database Tier

IBM Cloud offers managed database solutions including PostgreSQL and Db2. These services offer a low barrier of entry for users looking to deploy highly resilient and scalable database tiers. The Liferay OpenShift in IBM Cloud architecture has been tested with manually deployed containerized MariaDB and PostgreSQL servers and with IBM Cloud's PostgreSQL managed database solution.

Those looking to use other database platforms (e.g. Oracle, MS SQL Server, etc.) may rely on third-party Docker images instead of using a managed database service from IBM Cloud. You are free to choose any database platform, assuming the platform is supported according to the Liferay Support Matrix.

Please see the Liferay DXP Compatibility Matrix for a complete list of supported database platforms.

Search Tier

For this tier, we need to set up an Elasticsearch (ES) cluster using a version that is compatible with DXP. See the Liferay DXP Compatibility Matrix to find the specific ES versions compatible with Liferay DXP 7.2.

Liferay DXP and Elastic recommend at least a three-node cluster so as to provide resiliency in case of random failures. Due to potential split-brain issues, it is not recommended to use a two-node cluster. For a three-node cluster, Elasticsearch recommends setting the `discovery.zen.minimum_master_nodes` property to 2. For more information, please consult the Elasticsearch node settings documentation.

You should use a similar JDK for running ES as used for running DXP. At minimum, the JDK major version should match and have the same vendor. This requirement stems from the fact that DXP uses the TCP endpoint of ES to communicate and in some edge cases, the JDK versions on each end may play a role in how the communication is handled.

For details on how to install and set up ES, please check the Elasticsearch documentation.

For testing the reference architecture, we specified a max of 4 (virtual) CPUs per ES node.

Liferay DXP Specific Considerations

File and Database Storage

Liferay DXP utilizes shared disk storage for its document management capabilities. In non-public cloud deployments, customers tend to provision network attached storage (NAS) or SAN drives mounted via NFS or similar technologies. IBM Cloud provides similar features.

Block and file storage options are both available for OpenShift on IBM Cloud. Each type of storage is available in various storage classes like bronze, silver, and gold, that differ in terms of size range and IOPS. See this article for details. Liferay DXP needs one storage volume for its Liferay Home folder, including the Liferay Documents and Media repository. It needs another storage volume for its database store.

If you're using IBM Cloud's PostgreSQL as a Service, you'll create your database storage there. On the other hand, if your database tier will be provided by a MariaDB or PostgreSQL Docker container, you'll need to create a database storage volume in the OpenShift Kubernetes cluster. Regardless of your database choice, you'll also need to create a file storage volume for the Liferay Home folder, including the Liferay Documents and Media repository. We recommend the `ibmc-file-retain-gold` storage class for Liferay Home and `ibmc-block-retain-gold` if you need to create a database volume.

Search

By default, Liferay DXP ships with an embedded Elasticsearch search engine. This means that the Elasticsearch engine runs in the same JVM as Liferay DXP. Although this solution is great for having out-of-the-box search in Liferay, it's not supported by Liferay for production use, only for development.

For production usage, Liferay only supports using Elasticsearch running outside of the DXP JVM (i.e., 1 JVM for Liferay DXP and a separate JVM for the Elasticsearch search engine).

Search engines benefit heavily from caching and their JVM memory profiles are substantially different from a JVM focused on serving content and web views (e.g. Liferay JVM). For these reasons, the two applications should always be kept separate in production environments.

While it's technically possible for the Elasticsearch JVM to run on the same IBM Cloud VM as the Liferay JVM, this will cause the two processes to compete for the same resources. For heavy search usage, Liferay strongly advises deploying not only to a separate process but to a IBM Cloud VM to provide dedicated CPU capacity. Thus, in the Liferay in OpenShift on IBM Cloud reference architecture, Elasticsearch is deployed independently.

Cloud Architecture Considerations

Security

The basic means of securing your Liferay system in IBM Cloud are security groups (as mentioned in the Firewall section above) and IAM (identity access management) features including users, service IDs, access groups, roles, policies, resources, resource groups, services, and service instances. See this article for an overview of IBM Cloud's IAM features.

SSL/TLS

In this reference architecture, the entrypoint for users into the Liferay in OpenShift on IBM Cloud system is an OpenShift route. This service offers SSL termination and can forward unencrypted traffic to the backend web or application servers. If you use an IBM-provided ingress subdomain, you can use an IBM-provided TLS certificate. If you use a custom domain, you can use your own TLS certificate to manage TLS termination. HAProxy is the default implementation of the routing layer of OpenShift. Thus, OpenShift routes support URL-based routing, allowing for more complex capabilities such as transparently utilizing a CDN for certain resources.

Some security requirements may require encryption on all communications including communication between the load balancer and the web tier or the web tier and the application tier. IBM Cloud's OpenShift route service allows you to re-encrypt packages before they're forwarded to upstream apps. See this article for more information.

DATA ENCRYPTION

IBM Cloud offers encryption of the data at rest and in transit for many of its data services. This includes, but is not limited to:

- Persistent storage encryption
- Bring your own key, enabled via a Kubernetes key management service (KMS) provider
- OpenShift worker node disk encryption
- OpenShift cluster secrets
- Data-in-use encryption

The list above is not exhaustive. Please see IBM Cloud's OpenShift documentation for more information about IBM Cloud's data encryption services in the context of an OpenShift cluster.

Autoscaling

Autoscaling is a common strategy used in cloud environments to improve fault tolerance and provide dynamic resource allocation. Since OpenShift is an abstraction on top of Kubernetes, OpenShift in IBM Cloud reaps all the benefits of the Kubernetes autoscaling and self-healing features. Pods and services will automatically start and stop based on the criteria you specify in your deployments.

IBM Cloud's OpenShift cluster autoscaler periodically scans the cluster and adjusts the number of worker nodes within its managed worker pools in response to workload resource requests and any custom settings, such as scanning intervals. Please refer to Autoscaling Clusters for detailed information about autoscaling IBM Cloud OpenShift clusters.

We will not discuss best practices and how to set up autoscaling in OpenShift. However, Kubernetes and OpenShift are designed to make autoscaling extremely easy. Please consult the appropriate OpenShift and Kubernetes documentation for guidelines. Liferay is fully compatible with autoscaling architectures, assuming you have an appropriate Liferay DXP subscription.

High Availability and Disaster Recovery

To support highly-available applications and deployments, IBM Cloud's OpenShift offering includes multiple, independent global server regions and zones. Both single-zone and multi-zone clusters are available. If you create a cluster in a multi-zone metro location, the replicas of your highly available

Kubernetes master are automatically spread across zones. If you create a cluster in a single zone (data center) location, you can create multiple worker nodes but you cannot spread them across zones. See this article for more information.

Multiple clusters can be set up across zones or regions and can be connected via global load balancers. Global load balancing services help ensure your Liferay system's resiliency against hardware or network interruptions. See this article for more information.

Note that some of IBM Cloud's OpenShift data storage solutions, such as file and block storage, are data center-specific and cannot be shared across zones in a multizone cluster setup. See this article for details. IBM Cloud's PostgreSQL as Service offering, on the other hand, offers a 99.99% uptime SLA, off site failover selection, backup services, and disaster recovery options.

AUTOMATED SETUP

To be prepared for various disaster scenarios, you should be able to recreate your entire Liferay DXP system, including the underlying system infrastructure and all its applications. This can be done by carefully documenting all of your infrastructure and application deployment steps. However, Liferay recommends fully automating the construction of your entire DXP system, as well as the on-going continuous deployment of applications being developed. With little to no manual intervention required, disaster recovery simply involves executing the automated infrastructure reconstruction scripts along with deploying the latest applications. This approach in building infrastructure automatically is known as "Infrastructure as Code" or "Configuration as Code". Various tools can help with automating the building of infrastructure and deployment scripts, such as Helm.

After a disaster, make sure to perform all successive updates (i.e., any updates that were performed after the initial installation) using your automated tools and scripts to make sure that your Liferay stack is recovered exactly to the state it was in before the failure.

BACKUP SCHEDULE AND REPLICATION

Properly backing up content and data is an important piece in planning for a successful data recovery procedure. Liferay DXP has three sets of data which need to be backed up and recovered after a failure:

- Database
- Document library file store
- Elasticsearch index

Please refer to IBM Cloud's documentation on backups for each of their services, such as IBM Cloud PostgreSQL as a Service and IBM Cloud file storage. As a best practice, plan on synchronizing the backups to all three sets of data, so that they remain as consistent as possible.

DATA RECOVERY

Please refer to IBM Cloud's documentation on backups and restoration steps for each of their services.

For Elasticsearch indexes, a restore of data from backup is not strictly necessary. The indexes can be reindexed on demand, although this process will take time depending on the amount of data to be indexed. Elasticsearch index backup and restore plans should take this into consideration.

PUTTING IT ALL TOGETHER

With both the scheduled backups and data recovery procedures in place, it is important to perform a disaster recovery test. The end goal of this test is to reconstruct a fully functional Liferay DXP system from the various backups, using the data recovery procedures. Of course, this should be done in an environment closely matching the production environment, but not the actual production environment (unless downtime is acceptable, or if the project has not been debuted). Additionally, make sure to regularly schedule disaster recovery tests. A disaster recovery plan that is not regularly tested is deficient.

Networking (Liferay clustering)

When Liferay DXP is deployed to OpenShift in IBM Cloud, you should enable virtual routing and forwarding (VRF) to move IP routing for your account and all of its resources into a separate routing table. If VRF is enabled, you can then enable IBM Cloud service endpoints to connect directly to resources without using the public network.

Liferay implements clustering with its Cluster Link feature. By default, Liferay Cluster Link uses multicast for discovery and communication between cluster members. In IBM Cloud, Cluster Link should be configured to use JDBC_Ping for member discovery and unicast for communication between cluster members. This is the setup we used for testing the reference architecture.

Cloud vs Bare Metal Performance

Special attention should be paid toward vocabulary used when describing IBM Cloud VM performance. With bare metal machines, a CPU refers to the physical chip which typically contains multiple cores. Each core may contain one or more hardware threads. With respect to the IBM Cloud platform, a core refers to a single physical hardware thread.

Keep in mind that virtualization entails a certain degree of overhead, compared to bare metal. For instance, while a b3c.8x32 might provide 66% of the threads from a physical CPU, it might actually offer less than 66% of bare metal performance. You should factor this into your calculation when performing capacity planning.

Summary

In the preceding sections, we outlined the steps and considerations to design a fully fault-tolerant Liferay DXP deployment for the IBM Cloud OpenShift platform. The described architecture builds a solid foundation for future growth.

Disclaimer

Liferay can only give you an initial tuning recommendation based on benchmarks that have been performed on the core Liferay DXP product. It is up to you as system architects and business analysts to come up with the utilization scenarios that your system will need to service the required amount of users.

It is your responsibility to run the appropriate load tests on your system before production deployment so that you can identify significant bottlenecks due to custom applications or portlets or other unforeseen system and network issues. Once you've run these tests, adjust your system and resource configurations to meet your performance requirements.

Please use this document and the Liferay DXP Deployment Checklist as guides in sizing your system and procuring your hardware.

Moving Forward

Liferay DXP Cloud

As an alternative to running Liferay in OpenShift on IBM Cloud, your team can consider Liferay DXP Cloud. DXP Cloud is a Platform as a Service (PaaS) tailored for Liferay DXP that will help you focus on what matters, saving IT resources for your highest business priorities. DXP Cloud helps teams focus on critical business needs and reduces the time that would otherwise be spent on infrastructure management. For more information, visit liferay.com/dxp-cloud or contact sales@liferay.com.

Liferay Global Services

Learn how Liferay's Global Services team can support your Liferay DXP project with a Go Live consultation. Contact sales@liferay.com for more information.

Liferay makes software that helps companies create digital experiences on web, mobile and connected devices. Our platform is open source, which makes it more reliable, innovative and secure. We try to leave a positive mark on the world through business and technology. Hundreds of organizations in financial services, healthcare, government, insurance, retail, manufacturing and multiple other industries use Liferay. Visit us at liferay.com.

200717