

Component	Purpose	AWS Service
<b>EKS Cluster</b>	Kubernetes orchestration platform for managing Pega pods	Amazon EKS
<b>Worker Nodes</b>	EC2 instances or Fargate tasks where Pega pods run	EC2 or AWS Fargate
<b>Pega Application Pods</b>	Containers running the Pega application	Amazon EKS
<b>Search &amp; Reporting Service (SRS)</b>	Indexing and search functionality	Amazon OpenSearch Service
<b>Batch &amp; Stream Nodes</b>	For background processing and data streaming	Amazon EKS
<b>Pega Database</b>	Relational database for storing Pega data	Amazon RDS or Amazon Aurora
<b>MSK (Managed Streaming for Kafka)</b>	Real-time data streaming and event-driven processing	Amazon MSK
<b>Data Storage</b>	Storage for static content, logs, and binary data	Amazon S3
<b>Search Service</b>	Full-text search and log analytics	Amazon OpenSearch Service
<b>Caching</b>	Caching frequently accessed data for performance improvement	Amazon ElastiCache (Redis or Memcached)
<b>VPC (Virtual Private Cloud)</b>	Isolate and secure infrastructure	Amazon VPC
<b>Load Balancers</b>	Distribute traffic across multiple Pega pods	AWS ALB/NLB
<b>Monitoring &amp; Logging</b>	Application performance monitoring and log management	AWS CloudWatch
<b>Secrets Management</b>	Secure storage and management of sensitive information	AWS Secrets Manager
<b>IAM (Identity &amp; Access Management)</b>	Managing permissions across AWS services	AWS IAM
<b>DNS Management</b>	Managing domain names and routing requests	Amazon Route 53
<b>File Storage</b>	Shared file storage accessible by multiple pods	Amazon EFS or Amazon FSx
<b>CI/CD Pipeline</b>	Automating deployment processes	AWS CodePipeline, CodeBuild, CodeDeploy

## EKS Components

=====

Web Server Pods  
Batch Pods  
Hazelcast STS  
Constellation pod  
Search and Reporting pod

External services

=====

RDS(Postgres)  
AWS MSK  
AWS openSearch  
ECR

=====

When deploying Pega 8.8 on Amazon EKS, the integration and dependencies between the Pega deployment components, Amazon OpenSearch, and Amazon MSK (Managed Streaming for Apache Kafka) are critical for the application's performance and reliability. Below is a breakdown of how these components interact and their dependencies:

### 1. Pega Application Pods

- **Role:** These pods run the core Pega application, handling user interactions, business logic, and process automation.
- **Dependencies:**
  - **Pega Database (RDS/Aurora):** The application pods rely on the database for storing and retrieving application data, configurations, and user session information.
  - **Amazon MSK:** Used for processing event streams, such as case processing, notifications, or real-time analytics.
  - **Amazon OpenSearch Service:** Supports search and reporting features by indexing data from Pega.

### 2. Search and Reporting Service (SRS)

- **Role:** Manages full-text search and reporting functionalities within Pega.
- **Dependencies:**
  - **Amazon OpenSearch:** Provides the search backend, indexing data from Pega for quick retrieval during searches and reports. SRS components within Pega use OpenSearch to store and query large volumes of data efficiently.
  - **Pega Database:** Acts as the source for indexing data. Any updates in the database may trigger re-indexing in OpenSearch.

### 3. Batch and Stream Nodes

- **Role:** Handles background processing, such as asynchronous jobs, scheduled tasks, and data streaming.
- **Dependencies:**

- **Amazon MSK:** Batch and Stream nodes often consume messages from Kafka topics, process the data, and produce results back to other Kafka topics or persist them in the Pega database.
- **Pega Database:** Stores processed results and intermediate data. Batch processing might involve large data sets requiring efficient read/write operations.
- **Amazon OpenSearch:** Can be used for storing the results of certain batch operations that need to be quickly searchable or analyzable.

#### 4. Constellation Pod

- **Role:** Manages real-time UI rendering and front-end operations in a Pega deployment.
- **Dependencies:**
  - **Pega Application Pods:** Interacts with the main Pega application to retrieve and update data.
  - **Amazon MSK:** In some scenarios, Constellation Pods might interact with MSK for real-time updates or notifications, enabling dynamic UI changes based on Kafka event streams.
  - **Amazon OpenSearch:** If search capabilities are embedded within the UI, the Constellation Pod will query OpenSearch to provide results to the end-users.

#### 5. Hazelcast

- **Role:** A distributed in-memory data grid, used by Pega for caching and enhancing performance.
- **Dependencies:**
  - **Pega Application Pods:** Heavily relies on Hazelcast for caching frequently accessed data, reducing the load on the database and other persistent storage solutions.
  - **Amazon MSK:** Hazelcast might use MSK for pub/sub messaging patterns in complex distributed cache scenarios.
  - **Amazon OpenSearch:** While Hazelcast and OpenSearch are distinct, Hazelcast might be used to cache search results, reducing the frequency of repeated queries to OpenSearch.

#### 6. Integration with Amazon MSK

- **Role:** Amazon MSK is used for real-time data streaming, enabling event-driven architectures within Pega.
- **Dependencies:**
  - **Batch and Stream Nodes:** These nodes consume and produce messages in Kafka, enabling asynchronous processing and real-time data updates.
  - **Pega Application Pods:** Directly interact with MSK for event-driven tasks, such as triggering case actions based on events or integrating with external systems.
  - **Search and Reporting Service:** May use Kafka streams to index data continuously as it arrives, improving the freshness of the data in OpenSearch.

#### 7. Integration with Amazon OpenSearch

- **Role:** Amazon OpenSearch handles full-text search, log analytics, and reporting for Pega.
- **Dependencies:**
  - **Search and Reporting Service (SRS):** Directly depends on OpenSearch for indexing and querying data. The SRS component needs OpenSearch to be available and responsive to ensure search operations in Pega work efficiently.

- **Pega Application Pods:** Use OpenSearch for complex searches and reporting tasks. They may send logs and other operational data to OpenSearch for monitoring and analytics.
- **Batch and Stream Nodes:** Might push processed data to OpenSearch for further analysis or querying.

## Conclusion

The integration of Pega components with Amazon MSK and OpenSearch creates a powerful, scalable architecture that supports real-time processing, complex searches, and efficient background processing. Ensuring that these dependencies are correctly configured and optimized is crucial for the overall performance and reliability of your Pega deployment on AWS EKS.

---

---

### Pega components:

<https://github.com/pegasystems/pega-helm-charts/blob/master/charts/pega/README.md>

<https://github.com/pegasystems/pega-helm-charts/blob/master/charts/addons/README.md>

<https://github.com/pegasystems/pega-helm-charts/blob/master/charts/backingservices/README.md>

\$ helm repo add pega <https://pegasystems.github.io/pega-helm-charts>

### Understanding Pega Architecture

<https://docs.pega.com/bundle/platform-88/page/platform/deployment/client-managed-cloud/pega-kubernetes-architecture.html#cmc-deployment-architecture>

<https://docs-previous.pega.com/client-managed-cloud/83/understanding-pega-deployment-architecture>

<https://docs.pega.com/bundle/platform-88/page/platform/deployment/platform-support-guide/platform-support-guide.html>

<https://docs.pega.com/bundle/platform-88/page/platform/deployment/client-managed-cloud/requirements-prerequisites.html>

<https://docs.pega.com/bundle/platform-88/page/platform/deployment/externalization-of-services/externalization-of-services.html>

<https://docs.pega.com/bundle/platform-88/page/platform/deployment/externalization-of-services/externalize-search-in-your-deployment.html>

### Constellation service deployment

<https://docs.pega.com/bundle/platform-88/page/platform/deployment/constellation/constellation-overview.html>

## Docker Images:

<https://docs-previous.pegacorp.com/client-managed-cloud/83/using-pegacorp-provided-docker-images>

<https://docs.pegacorp.com/bundle/platform-88/page/platform/deployment/client-managed-cloud/pegacorp-docker-images-manage.html#cmc-docker-images-overview>

## Elastic Search:

Note: SRS can connect to the Elasticsearch 7.10 version running on AWS OpenSearch service, but it must be configured to use the legacy Elasticsearch OSS engine, version 7.10. However, SRS may not work as expected with OpenSearch search database, nor with AWS OpenSearch service that is running OpenSearch 1.x or 2.x managed versions.

<https://docs.pegacorp.com/bundle/platform-88/page/platform/deployment/externalization-of-services/deploying-srs.html#tsk-deploying-srs>

<https://docs.pegacorp.com/bundle/platform-88/page/platform/deployment/externalization-of-services/externalize-search-in-your-deployment.html>

Questions on ES and STS and EKS

Their Versions, Authentications, Opensearch Compatibility

## HazelCast:

<https://docs.pegacorp.com/bundle/platform-88/page/platform/deployment/externalization-of-services/tsk-externalize-hazelcast-in-your-deployment.html>

```
$ kubectl create namespace mypega
```

```
$ kubectl create namespace pegaaddons
```

```
$ kubectl create namespace pegabackingservices
```

```
$ helm install backingservices pega/backingservices --namespace pegabackingservices --values backingservices.yaml
```

```
$ helm install addons pega/addons --namespace pegaaddons --values addons.yaml
```

## K8S Architecture

<https://docs.pega.com/bundle/platform-88/page/platform/deployment/externalization-of-services/externalization-of-services.html>

Client Managed Cloud (8.8)  
Deployment Overview

