OKV for EDB Postgres TDE

How to integrate PostgreSQL with OKV

May, 2025 Version 1.1

Copyright © 2025, Oracle and/or its affiliates

Public

Purpose statement

This document provides a step by step guide how to integrate OKV with EDB Postgres TDE for key management.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of contents

[Purpose 4](#_Toc198288143)

[Introduction 4](#_Toc198288144)

[Required software 4](#_Toc198288145)

[How does the integration work 4](#_Toc198288146)

[How to setup 4](#_Toc198288147)

[Initdb 4](#_Toc198288148)

[Start and create the database 5](#_Toc198288149)

# Purpose

This document provides a step by step guide how to integrate OKV with EDB Postgres TDE for key management.

# Introduction

EDB Postgres TDE uses a Key Encryption Key that gets stored in a file key.bin in the tde\_encryption directory of the database. During the initdb phase this key can be wrapped using a key encryption key from a KMIP compliant KMS.  
For the integration with Oracle Key Vault a java class has been created using the software development kit.

# Required software

In my setup I used the following software and versions

* Oracle Linux 8 server for the Database install
* EDB Postgres Advanced Server version 17.5
* Oracle Key Vault 21.10
* Software Development Kit of OKV
* JDK 1.8

# How does the integration work

During the initdb phase of EDB the keymanagement commands for “key wrap” and “key unwrap” are passed. This can be achieved either by specifying two environment variables or by passing commandline arguments.

The key wrap command will be used to wrap the key generated by initdb before writing it into the pg\_encryption/key.bin file located in the database directory. The key unwrap command will also be used during initdb and will be added to the file postgresql.conf as entry data\_encryption\_key\_unwrwap\_command.

The key wrap and key unwrap commands are shell scripts that call the java class with the correct arguments.

# How to setup

Unpack kmip.tar in the HOME directory of the user enterprisedb. The system used for the test uses /var/lib/edb.

On OKV create a new Endpoint (type other) with a default wallet and in that wallet create a non-extractable symmetric key. The Unique ID of the new key will be used during the initdb command.

In the usual way deploy the endpoint into /var/lib/edb/kmip/okv directory.

# Initdb

Test you java version before starting initdb to be 1.8.

If the kmip.tar file was not extracted in /var/lib/edb, please update KMIP\_DIR entries in encrypt.sh and decrypt.sh files in the kmip/bin directory to reflect the correct path.

For the purpose of this document the initdb will be run from /var/lib/edb as follows:

initdb -D hrprod -y --key-unwrap-command='/var/lib/edb/kmip/bin/decrypt.sh 7CD1A23D-AEC4-4D25-9333-BDDE6B045245 "%p"' --key-wrap-command='/var/lib/edb/kmip/bin/encrypt.sh "%p" 7CD1A23D-AEC4-4D25-9333-BDDE6B045245' -U admin -W --auth-local=trust

In this command hrprod is the name of the database and 7CD1A23D-AEC4-4D25-9333-BDDE6B045245 the unique ID of the key in OKV.

# Start and create the database

After initdb successfully finishes start the database:

$ pg\_ctl -D hrprod -l logfile start

and create the database:

$ createdb hrprod -U admin

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

[](https://blogs.oracle.com/)[](https://www.facebook.com/Oracle/)[](https://twitter.com/oracle) blogs.oracle.com facebook.com/oracle twitter.com/oracle

Copyright © 2025, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.