

User Guide

FORTANIX CONFIDENTIAL COMPUTING MANAGER – ADD AND EDIT AN APPLICATION

VERSION 5.0

TABLE OF CONTENTS

1.0	INTRODUCTION	2
2.0	CONTACT INFORMATION	3
3.0	DESCRIPTION OF SERVICES	3
3.1	Fortanix Confidential Computing Manager	3
3.2	Intel® SGX	3
3.3	Intel Attestation and Why it is Required	3
3.4	Navigation Buttons	4
4.0	ADD AND EDIT AN APPLICATION	5
4.1	Add Enclave OS Application	6
4.1.1	Enclave OS Directory/Filesystem Protections.....	14
4.1.2	Edit an Enclave OS Application.....	15
4.2	Add EDP Application	19
4.2.1	Edit an EDP Application.....	22
4.3	Add ACI Application	23
4.3.1	Edit an ACI Application.....	24
4.4	Setting Environment Variables for Your Application	26
5.0	DOCUMENT INFORMATION	27
5.1	Document Location	27
5.2	Document Updates	27

1.0 INTRODUCTION

Welcome to the Fortanix Confidential Computing Manager (CCM) User Guide. This document describes the steps to add and edit an application in the Fortanix Confidential Computing Manager. The users are provided the ability to quickly and easily navigate the interface to run containerized applications accordingly.

A Fortanix Confidential Computing Manager (CCM) Application is a program or service that is being protected with Runtime Encryption. In a microservice architecture, you might create an application in CCM for each of your microservices.

Since the code for an application is typically updated over time, an application definition in CCM is not associated with a particular enclave hash (MRENCLAVE). Instead, an application can be associated with one or more images, each of which represents a specific version of the application. MRENCLAVE values are associated with images.

The application record in CCM defines general characteristics for the application including the domain name(s) assigned to the application and, if the application is using EnclaveOS, the parameters to use when processing the application in the EnclaveOS converter. In the future, CCM will allow defining policies indicating where the application can run and what other applications and data stores it can communicate with.

DOCUMENT IDENTIFICATION INFORMATION

DOCUMENT NAME	GUIDE, USER, CONFIDENTIAL COMPUTING MANAGER
DATE CREATED	14 MAY 2020
SECURITY CLASSIFICATION	For use by Fortanix internal and Fortanix Confidential Computing Manager Customers ONLY.

2.0 CONTACT INFORMATION

CONTACT INFORMATION		
ITEM	PRIMARY	ALTERNATE
NAME	Fortanix	
EMAIL ADDRESS	Fortanix Support Link	
CONTACT NUMBER	N/A	
TITLE	N/A	
SUPPORT HOURS	8am - 5pm Monday - Friday	

3.0 DESCRIPTION OF SERVICES

3.1 FORTANIX CONFIDENTIAL COMPUTING MANAGER

Fortanix Confidential Computing Manager provides “data-in-use” protection for your container workloads. It leverages the Intel® Software Guard Extensions (SGX) technology to run code and data in CPU-hardened “enclaves” or a “Trusted Execution Environment” (TEE). The enclave is a trusted area of memory where critical aspects of the application functionality are protected, helping keep code and data confidential and unmodified.

3.2 INTEL® SGX

Intel® SGX is an extension to the x86 architecture that allows running applications in a completely isolated secure enclave. The application is not only isolated from other applications running on the same system, but also from the Operating System and possible Hypervisor. This prevents administrators from tampering with the application once it is started. The memory of secure enclaves is also encrypted to thwart physical attacks.

The technology also supports storing persistent data securely such that it can only be read by the secure enclave. In addition, you can prove remotely that your application is running in a secure enclave using remote attestation.




3.3 INTEL ATTESTATION AND WHY IT IS REQUIRED




Since enclaves are instantiated on platforms by untrusted code, before enclaves are provisioned with application confidential information, it is essential to be able to confirm that the desired enclave was correctly instantiated on a platform protected by Intel SGX. This is done by a remote

attestation process. Remote attestation consists of using Intel SGX instructions and platform software to generate a “quote” that combines the enclave digest with a digest of relevant enclave data and a platform-unique asymmetric key into a data structure that is sent to a remote server over an authenticated channel. If the remote server concludes that the enclave was instantiated as intended and is running on a genuine Intel SGX-capable processor, it will provision the enclave as required.

3.4 NAVIGATION BUTTONS

The Navigation buttons for Fortanix Confidential Computing Manager are located on the left panel of the GUI and identify the screen functionality. The following table illustrates button functions:

NAVIGATION BUTTONS	
MENU LIST	FUNCTIONALITY
 INFRASTRUCTURE	<p>Click this menu item to see:</p> <ul style="list-style-type: none"> All the Compute Nodes that are part of your cluster. You can view SGX software version, secure application’s information, and attestation status of each of these Compute Nodes on which your Fortanix Confidential Computing Manager components are running. All the Compute Clusters that you have configured in Fortanix Confidential Computing Manager.
 GROUP	<p>Click this menu item to create a group, which is a collection of users and objects. A group helps users to manage identities and create third-party groups. It also helps in organizing and securing applications, datasets, workflows, and other resources that belong to the group.</p>
 APPLICATIONS	<p>Click this menu item to see:</p> <ul style="list-style-type: none"> All the Fortanix Confidential Computing Manager secured applications deployed on the cluster. An application is a way to let the service know which all parameters to configure for a Source

	<p>Container Image to run in SGX and where to push the converted Image.</p> <ul style="list-style-type: none"> All the Fortanix Confidential Computing Manager secured Docker images for the applications deployed on the cluster. All the application configurations used to customize the behavior for EDP/EnclaveOS applications.
 TASKS	<p>Click this menu item to see all the requests that need Administrator approval. For example, node enrolment, application domain approval, application image approval, and certificate issuance.</p>
 TOOLS	<p>Click this menu item to access the SGX Converter tool to convert an application.</p>
 USERS	<p>Click this menu item to see the list of users added to Fortanix Confidential Computing Manager. The Users page also allows you to edit the properties of a user and add new users.</p>

4.0 ADD AND EDIT AN APPLICATION

Prerequisites:

- A group must be created. *For more information, refer to [User's Guide: Create a Group](#).*
- Name of the input docker image of this application from the input registry.
- Output image location.

Steps to Add an Application:

You can convert, deploy, and approve your application all at the same time using Fortanix Confidential Computing Manager.

- Sign in to the Fortanix Confidential Computing Manager user interface, click the **Applications** menu item from the CCM UI left navigation bar.
- On the **Applications** page, click **+ ADD APPLICATION** to add a new application.

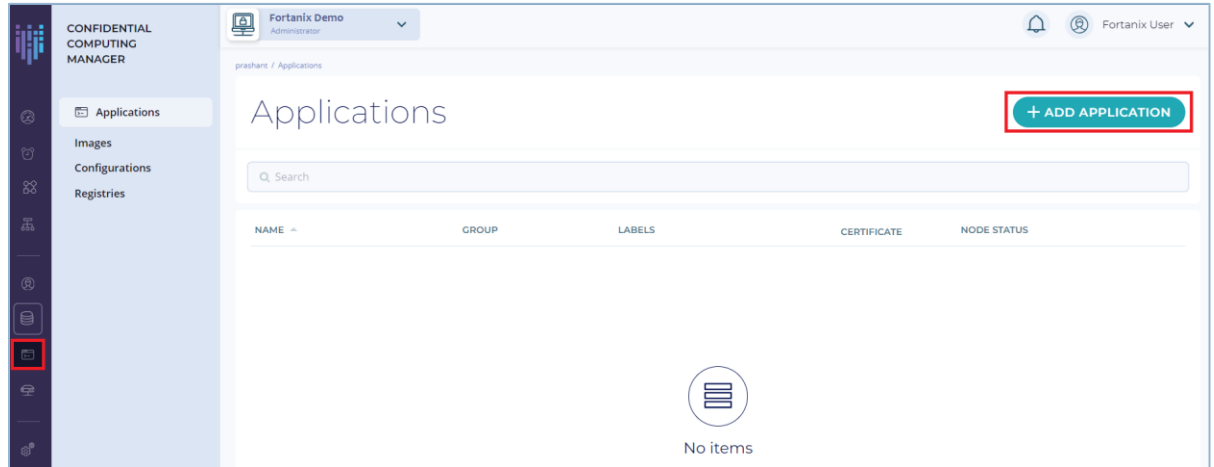


FIGURE 1: ADD NEW APPLICATION

3. There are three types of applications that you can add:
 - a. Add **EDP Application**
 - b. Add **Enclave OS Application**
 - c. Add **ACI Application**

4.1 ADD ENCLAVE OS APPLICATION

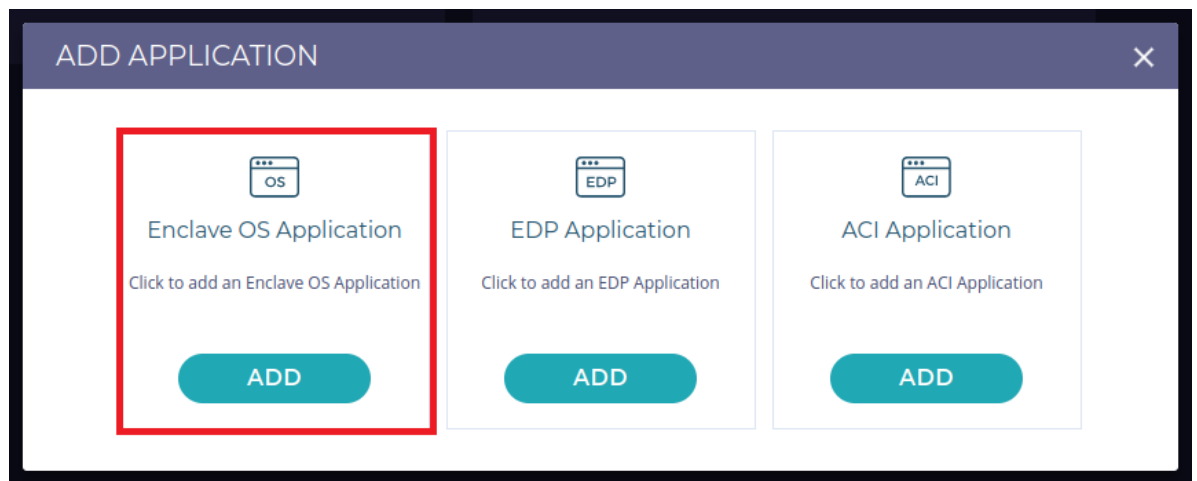


FIGURE 2: ADD ENCLAVE OS APPLICATION

1. In the **Enclave OS application** form, add a Flask Server Application. Fill in the following relevant details and click the **CREATE** button. You can use Fortanix's docker registry for the sample app.

Details:

Docker Hub: <https://hub.docker.com/u/fortanix/>

App: fortanix/python-flask

Optional: You can run the app with the following command:

```
sudo docker run fortanix/python-flask
```

```
fortanix-guest@em-vm1:~$  
fortanix-guest@em-vm1:~$ sudo docker run fortanix/python-flask  
* Serving Flask app "python" (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
* Debug mode: off  
* Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)
```

FIGURE 3: RUN THE APPLICATION



NOTE: It is recommended to use your private docker registry to keep the output image.

2. In the **Add application** form (**Figure 4**), fill in the following relevant details:
 - **Application name:** Enter the name of the application.
 - **Description** (optional): Enter the application's description.
 - **Input image name:** Enter the full path of the current application's current docker image.
 - **Output image name:** Enter the application's converted application name.
 - **Group:** Select the required group name from the drop down menu to associate the application with that Group.
 - **Add Labels:** To control which applications are allowed to run on which nodes, you need to add Labels for applications and nodes in the form of "Key:Value" pairs. *Refer to [Application and Compute Node Policy Enforcement](#) for more details.*
 - **Suggested Labels** – This field will show the top 10 labels that are frequently used by users of an account.
 - **Add Labels** – Enter the Key and Value pair and click the **LABEL** button to save the label. The newly created label will appear in the **Labels Added** field. You can also select an existing label from the **Suggested Labels** field.

Example of a “Key:Value” pairs is – “Location:Location_name” where “**Location**” is the Key and “Location_name” is the Value of the key such as “**South UK**”.

**NOTE:**

- A label’s key and value can have a maximum of 256 characters and is **case-sensitive**.
- Some keys are reserved for internal use which are called system defined labels.
 - Such as: 'Fortanix', 'fortanix', 'CCM', 'ccm, confidentialcomputingmanager. Or
 - {Fortanix|Fortanix|CCM|cm|confidentialcomputingmanager|Confidentialcomputingmanager}<Any_Non-Alphanumeric-Char> <Any-Char>.
- Adding labels for applications is not mandatory, even without labels applications can still run on the nodes. But if you are adding labels for an application then it is mandatory to add the same labels on the node on which the application will run.
- A node can have multiple labels that belong to different applications. For example:

App1’s label => Location1: Value1

App2’s label => Location2: Value2

Then the Node can have labels => Location1: Value1 , Location2: Value2.
- **Platform Configuration:** Fortanix Confidential Computing Manager allows you to run your confidential computing workloads on the following platforms:
 - Intel-SGX (including Azure DC-series VMs, and bare metal in the cloud or on-premises)
 - AWS Nitro

When the platform is Intel-SGX, expand the **Intel SGX** overlay menu in the **Platform Configuration** section, and enter the following values:

 - **ISVPRODID** is a numeric product identifier. You must select a unique value in the range of 0-65535 for their applications.
 - **Memory size** – Select the memory size from the drop down to change the memory size of the enclave.

- **Thread count** – Change the thread count to support the application.

When the platform is AWS Nitro, expand the **AWS Nitro Enclaves** overlay menu and enter the values for the following under **Enclave Parameters**:

- **Memory size** – Select the memory size from the drop down to change the memory size of the enclave.
- **CPU count** – Enter the number of vCPUs to allocate to the enclave. The number of vCPUs that you can allocate to an enclave depends on the size and configuration of the parent instance. If the parent instance is enabled for multithreading, you must leave at least 2 vCPUs for the parent instance. If multithreading is not enabled, you must leave at least 1 vCPU for the parent instance. For example, if your parent instance has 4 vCPUs and it is enabled for multithreading, you can allocate up to 2 vCPUs to the enclave.
- **File persistence** – This option is selected by default. This feature allows you to save the filesystem changes to an encrypted container mount. It allows the Nitro system to access a managed security object in Fortanix DSM to be able to encrypt and decrypt the Linux Unified Key Setup (LUKS) overlay file system. *For more details, refer to [User's Guide: AWS Nitro File Persistence](#).*



NOTE: For the File Persistence feature to work, you **must** configure the app certificate as described below, since when a Nitro image runs, it must be configured ahead of time to receive a certificate, which will authorize access to Fortanix DSM to obtain the keys for the Linux Unified Key Setup (LUKS) volume. Without the app certificate, this feature will not work.

- **Certificate Configuration:** Add any certificate using the **Certificate Configuration** section. A converted application can request a certificate from the Fortanix Confidential Computing Manager when your application is started. The certificates are signed by the Fortanix Confidential Computing Manager Certificate Authority, which issues certificates only to enclaves presenting a valid attestation.
 - **Domain:** Enter the allowed domain for the application. This is the domain that appears in the TLS certificate issued by the Fortanix Confidential Computing Manager.

- **Key path:** Enter the key path that will be accessible by the application.
- **Key type:** Select the type of key from the drop down menu that you want to generate.
- **Certificate path:** Enter the certificate path that will be accessible by the application.
- **RSA Key Size:** Select the size of the RSA keys in bits from the drop down menu.
- **Chain path** (optional): Enter the chain path for the complete certificate chain.
- Edit any **Advanced settings** that you might want to change.
 - **Environment variables** – Enter any environment variables that will be set at runtime. The variables need to be comma separated values.
 - **Read/Write directories** – Enter comma separated absolute paths of file system directories to allow read/write by the application, without encryption or integrity protection. Use this only if you understand the security implications. *For more details refer to the Section 4.1.1: Directory Protection for Enclave OS Applications.*
 - **Java runtime** – Select the appropriate Java runtime values. When you select the Java Runtime option for an application, the converted docker image will run with the specified options for the chosen JVM (Java Virtual Machine).

```

OPENJDK / ORACLE -
-XX:CompressedClassSpaceSize=16m
-XX:-UsePerfData
-XX:ReservedCodeCacheSize=16m
-XX:-UseCompiler
-XX:+UseSerialGC
OPENJ9 / LIBERTY -
-Xnojit
-Xnoaot
-Xdump:none
  
```

- **CA Cert Path** – Enter the path to store the Fortanix Confidential Computing Manager CA certificate.

As an optional step, you can install the CA certificate in the system trust store where all the certificates are stored. The following are the three options given:

- **Yes, install and continue image conversion even if the installation fails** – select this option if you want to convert the image even after the CA Certificate installation fails.
- **Yes, install and fail image conversion if the installation fails** – select this option if you want to stop image conversion after the CA Certificate installation fails.
- **No, do not install** – select this option if you do not want to install the CA Certificate.

CONFIDENTIAL COMPUTING MANAGER

Fortanix Demo

prashant / Applications / View application

Add application

Add the details of an application which will be deployed in the cluster. These attributes will be used to create secure images of the application which will eventually get deployed on the cluster.

Application name Python Application Server

Description (optional)

Input image name docker.io/fortanix-private/python-flask

Output image name docker.io/fortanix-private/python-flask-sgx

Group Select group

Add Labels

Suggested Labels

key3: value3 key2: value2 key1: value1 location: UK South

Add Labels

Enter key Enter value **ADD LABEL**

Please fill in this field

Added Labels

No Labels Added

Platform Configuration

Intel SGX

Enclave Parameters

ISVPRODID 1 **Memory size** 1 GB

Thread count 128

The screenshot displays the configuration interface for AWS Nitro Enclaves. It is divided into several sections:

- Enclave Parameters:** Includes fields for Memory size (set to 1 GB) and CPU count (set to 128). A checkbox for File persistence is checked, with a link to Read documentation.
- Certificate Configuration:** A modal window showing Domain (www.example.com), Type (Certificate Issued by Confidential Computing Manager), Key type (RSA), RSA Key Size (Choose size), Key path, Certificate path, and Chain path.
- ADVANCED SETTINGS:** Includes Environment variables (HOST=1.1.1.1, DEBUG=true), Encrypted directories (/tmp, /run), Read/Write directories (/app/logs, /app/tmp), Java runtime (Select), CA Cert Path, and options to install the CA Certificate into the system trust store.

At the bottom right, there are CANCEL and CREATE buttons. The footer contains the version 3.05.1838 and a Submit feedback link.

FIGURE 4: APPLICATION DETAILS

- Click the **CREATE** button to configure the image. The application will now be deployed and added to your approval and visible in the **APPLICATION** tab. You can approve the approval request in the **Tasks** tab.

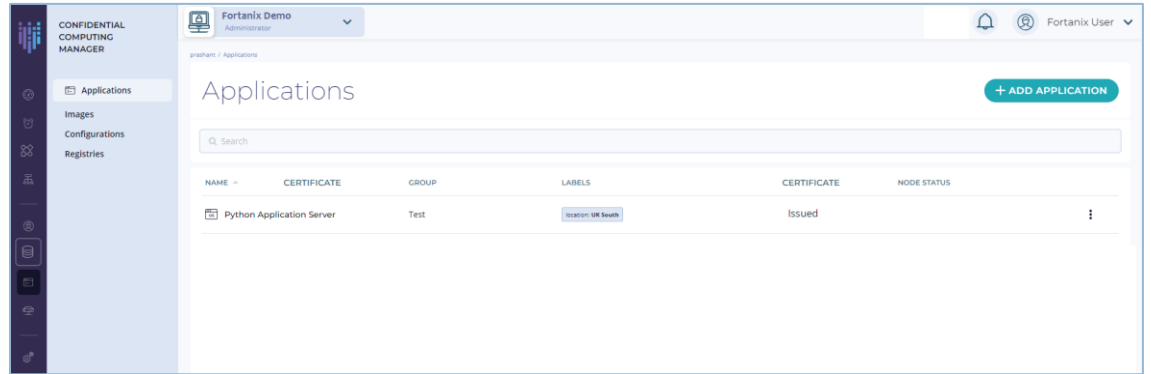


FIGURE 5: APPLICATION CREATED



NOTE:

- Creating an application does not mean that an SGX Ready Image is created and pushed. An application will be converted and pushed to the specified location once an image of this application is created.
- It is also possible to add labels for an Enclave OS application from the detailed view of an application.

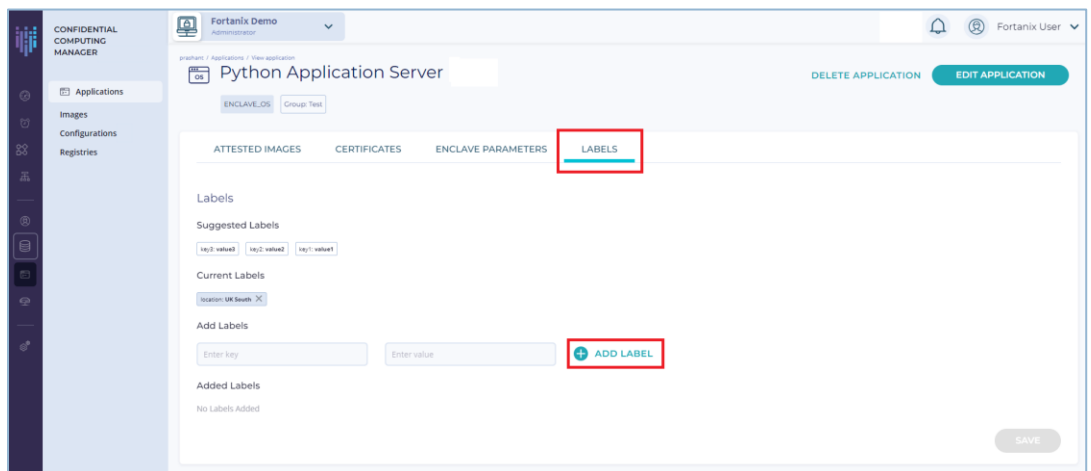


FIGURE 6: ADD LABELS

Refer to [User's Guide: Create an Image](#) to create an image for the Enclave OS application.

4.1.1 ENCLAVE OS DIRECTORY/FILESYSTEM PROTECTIONS

Enclave OS (EOS) provides file system integrity protection. There are three possible directory configurations within an Enclave:

- **Read-only** (integrity protected, not encrypted, not writable) – This is the default configuration.

- **Encrypted** (integrity protected, encrypted, but initial contents are unencrypted).
- **Read-write** (unprotected).

For files in **read-only** directories, if Enclave OS detects that a file has been modified, it will halt the execution of the Enclave. Enclave OS will ensure that the complete root tree (all directors below "/") have **read-only** permission, except for the following directories: `/etc`, `/run`, `/tmp`, `/opt/fortanix/enclave-os/app-config/rw/`, since these directories have **read-write** permissions. Except for `/etc`, the other directories are **encrypted** to prevent potential tamperers from outside the Enclave.



NOTE: At the time of Enclave OS application creation, you can configure additional directories to have **read-write** permissions.

Let us examine a typical use case:

An enclaved Python Flask application will load `myapp.py` file when the enclave starts up. If this file was in a **read-only** folder and it was modified outside of the enclave, at run-time when the file is loaded by Flask, Enclave OS will detect the tamper and halt execution. If the `myapp.py` file was in the **encrypted** folder but modified from outside the enclave, it will detect the tamper and halt execution.



NOTE: If you are using the API to create the app, the **read-write** directories can be specified in the JSON. For example:

```
"rw_dirs": ["/var/cache/nginx", "/etc/ssl"]
```

4.1.2 EDIT AN ENCLAVE OS APPLICATION

You can edit an application after you add it to your list.

1. Sign in to Fortanix Confidential Computing Manager, and then navigate to the **APPLICATION** menu item in the CCM UI left navigation bar.
2. Click the name of the application that you want to edit. A new screen opens where you can review and edit the configuration including certificates and deployed images.
3. Click the **EDIT APPLICATION** button.

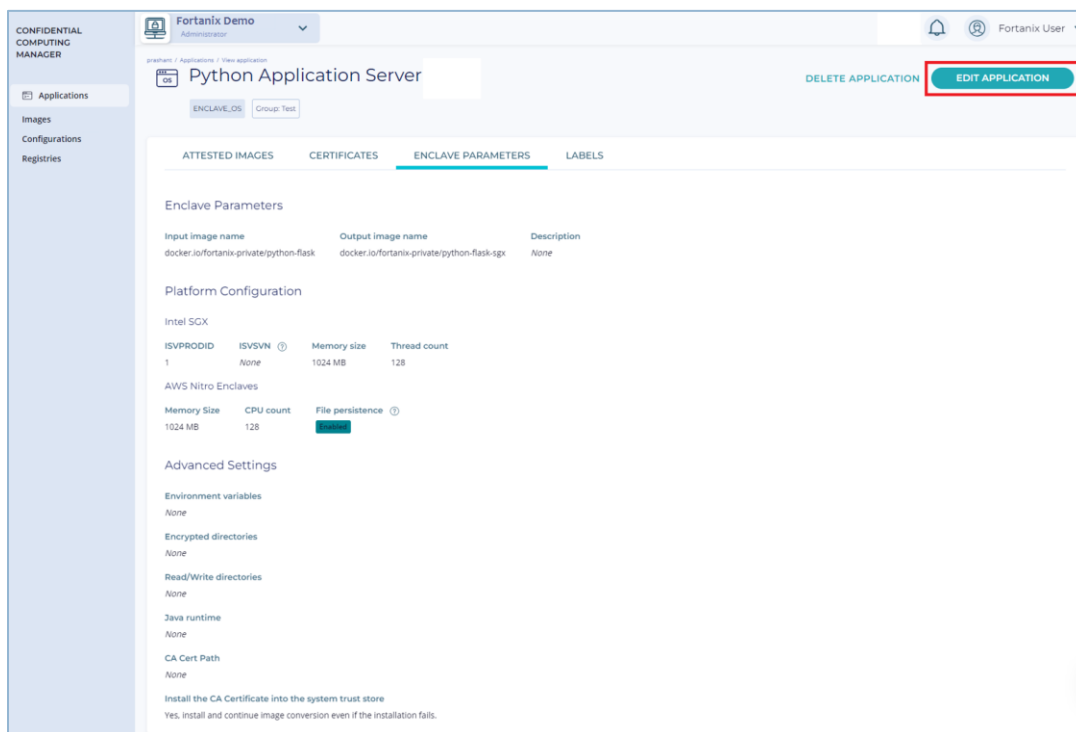


FIGURE 7: EDIT APPLICATION

4. Update the configuration that you want to make. Ensure that you understand the way that changing the advanced settings affects your application before you make any changes.
5. Click the **UPDATE** button.

CONFIDENTIAL COMPUTING MANAGER

Fortanix Demo

Applications / Edit application

Edit application

Edit the attributes of an application which will be used to create secure images of the application which will eventually get deployed on the cluster. These edits will apply to future images only.

Application name: Python Application Server

Description (optional):

Input image name: docker.io/fortanix-private/python-flask

Output image name: docker.io/fortanix-private/python-flask-sgx

Group: Test

Add Labels

Suggested Labels

key1=value1 key2=value2 key3=value3

Current Labels

location: UK South X

Add Labels

Enter key Enter value + ADD LABEL

Added Labels

No Labels Added

Platform Configuration

Intel SGX

Enclave Parameters

ISVPRODID: 1 Memory size: 1 GB

Thread count: 128

FIGURE 8: UPDATE APPLICATION



NOTE:

- The **Application name** cannot be edited.
- **Allowed domain** can only be edited if the application does not have any pending domain approval tasks.

4.2 ADD EDP APPLICATION

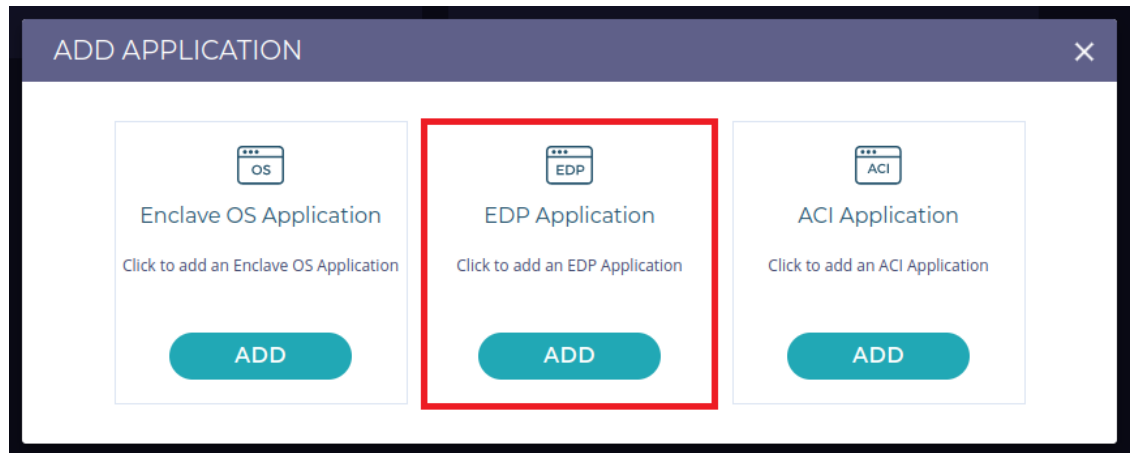


FIGURE 9: ADD EDP APPLICATION

1. In the EDP **Add application** form, fill in the following relevant details:
 - **Application name:** Enter the name of the application.
 - **Description** (optional): Enter the application's description.
 - **Group:** Select the required group name from the drop down menu to associate the application with that Group.
 - **Add Labels:** To control which applications are allowed to run on which nodes, you need to add Labels for applications and nodes in the form of "Key:Value" pairs. *Refer to [Application and Compute Node Policy Enforcement](#) for more details.*
 - **Suggested Labels** – This field will show the top 10 labels that are frequently used by users of an account.
 - **Add Labels** – Enter the Key and Value pair and click the **LABEL** button to save the label. The newly created label will appear in the **Labels Added** field. You can also select an existing label from the **Suggested Labels** field.

 **NOTE:**

- A label's key and value can have a maximum of 256 characters and is **case-sensitive**.
- Some keys are reserved for internal use which are called system-defined labels.
 - Such as: 'Fortanix', 'fortanix', 'CCM', 'ccm', confidentialcomputingmanager. Or

- {Fortanix|Fortanix|CCM|ccm|confidentialcomputingmanager|Confidentialcomputingmanager}<Any_Non-Alphanumeric-Char><Any-Char>.
- Adding labels for applications is not mandatory, even without labels applications can still run on the nodes. But if you are adding labels for an application then it is mandatory to add the same labels on the node on which the application will run.
- A node can have multiple labels that belong to different applications. For example:
App1's label => Location1: Value1
App2's label => Location2: Value2
Then the Node can have labels => Location1: Value1, Location2: Value2
Example of a "Key:Value" pairs is – "Location:Location_name" where "**Location**" is the Key and "Location_name" is the Value of the key such as "**South UK**".
- **Certificate Configuration:** Add any certificate using the **Certificate Configuration** section. The em-app RUST library can be used by EDP apps to obtain a signed CCM Certificate over enclave-generated certificates. You can select to add multiple certificates using the **ADD A CERTIFICATE** button.
 - **Domain:** Enter the allowed domain for the application. This is the domain that appears in the TLS certificate issued by the Fortanix Confidential Computing Manager.
 - **Type:** Enter the type of certificate to obtain for the application.

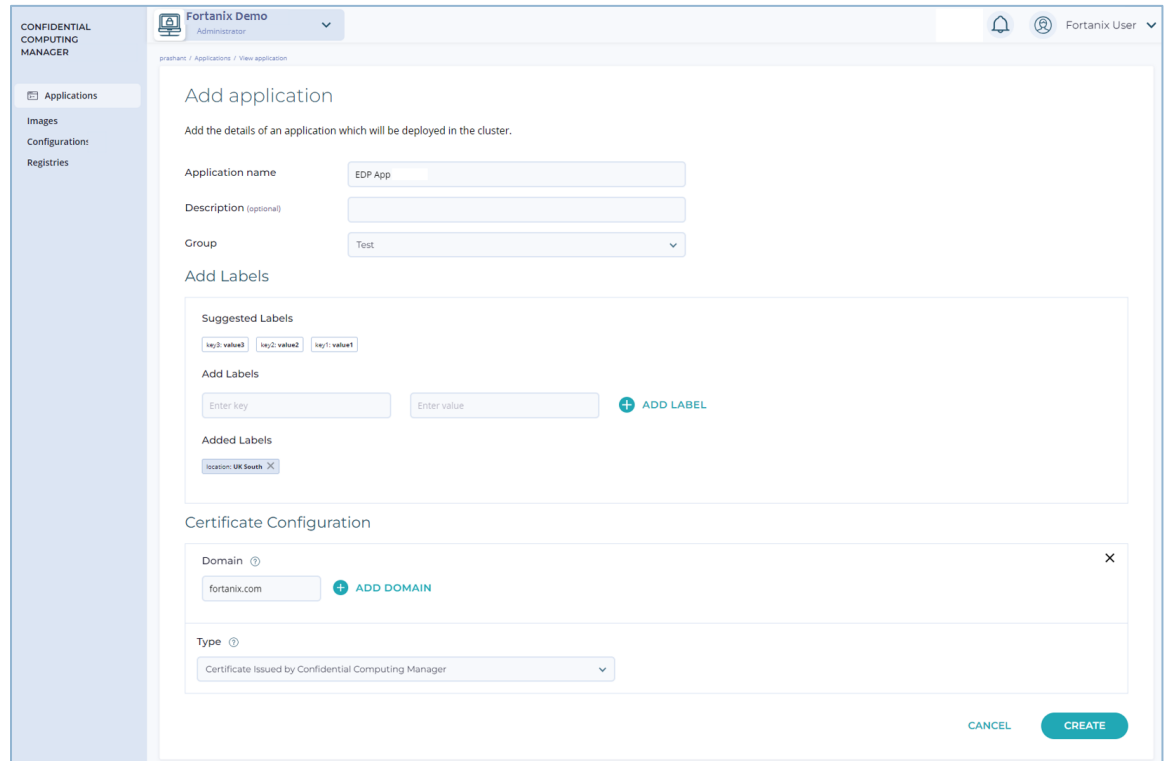


FIGURE 10: ADD EDP APPLICATION DETAILS

- After you configure all the certificates, click the **CREATE** button to configure the application.



NOTE: It is also possible to add labels for an application from the detailed view of the EDP application.

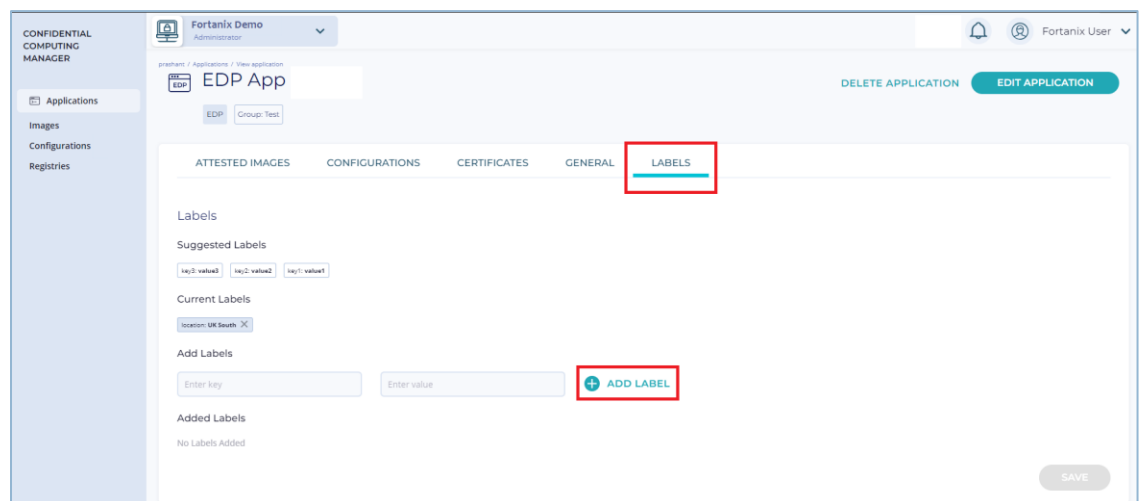


FIGURE 11: ADD LABELS

Refer to [User's Guide: Create an Image](#) to create an image for the EDP application.

4.2.1 EDIT AN EDP APPLICATION

You can edit an EDP application after you add it to your list.

1. Sign in to Fortanix Co, and then click the **APPLICATION** menu item in the CCM UI left navigation bar.
2. Click the name of the application that you want to edit. A new screen opens where you can review and edit the configuration including certificates.

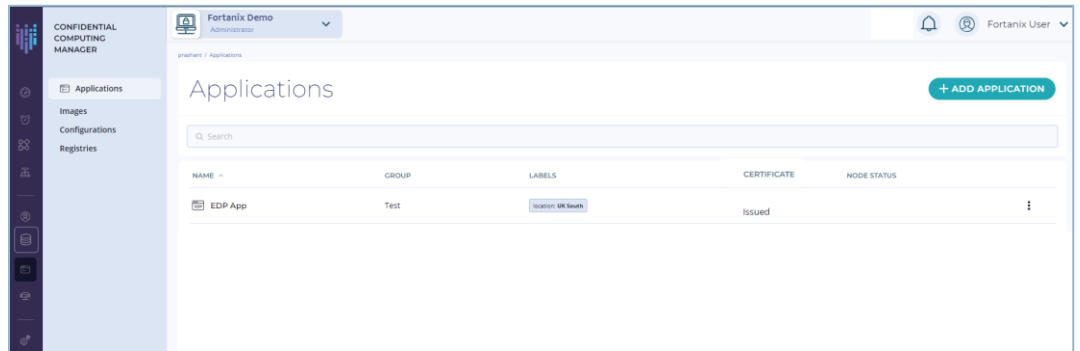


FIGURE 12: EDIT EDP APPLICATION

3. Click **EDIT APPLICATION**.

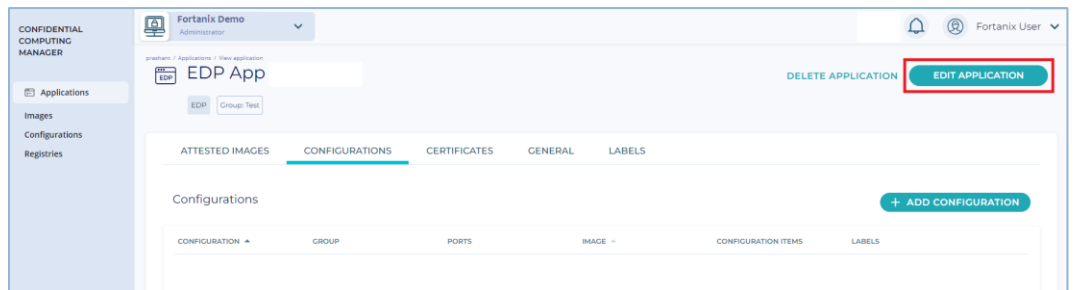


FIGURE 13: EDIT EDP APPLICATION

4. Update the configuration that you want to make.
5. Click the **UPDATE** button.

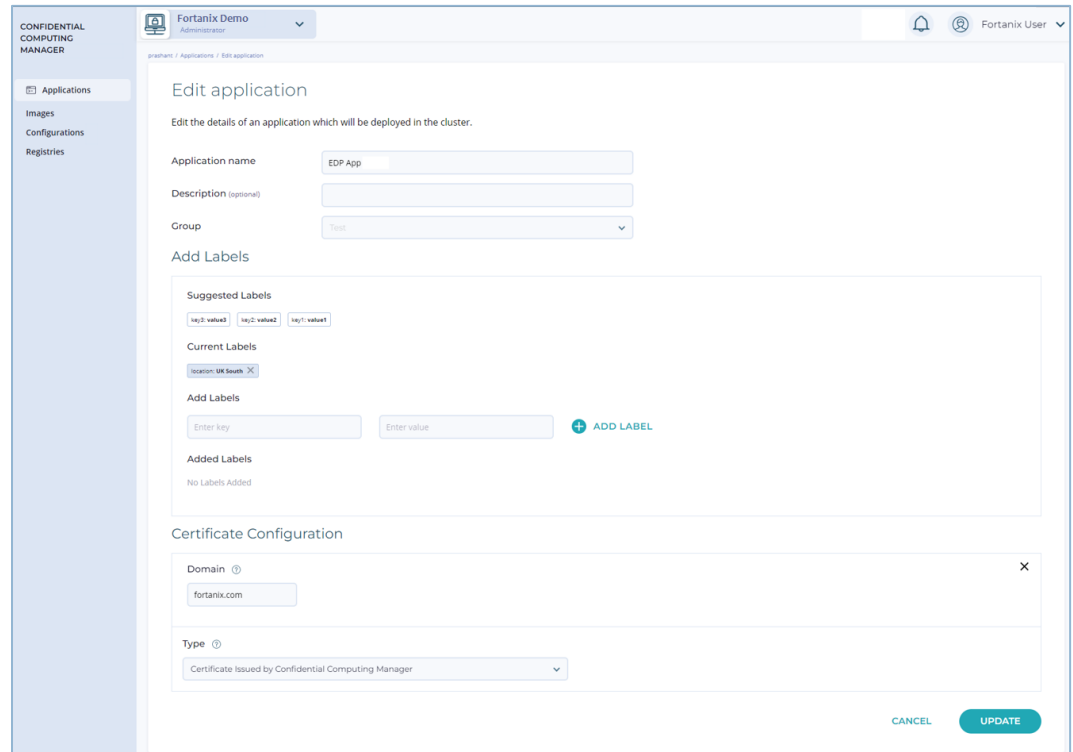


FIGURE 14: UPDATE APPLICATION

4.3 ADD ACI APPLICATION

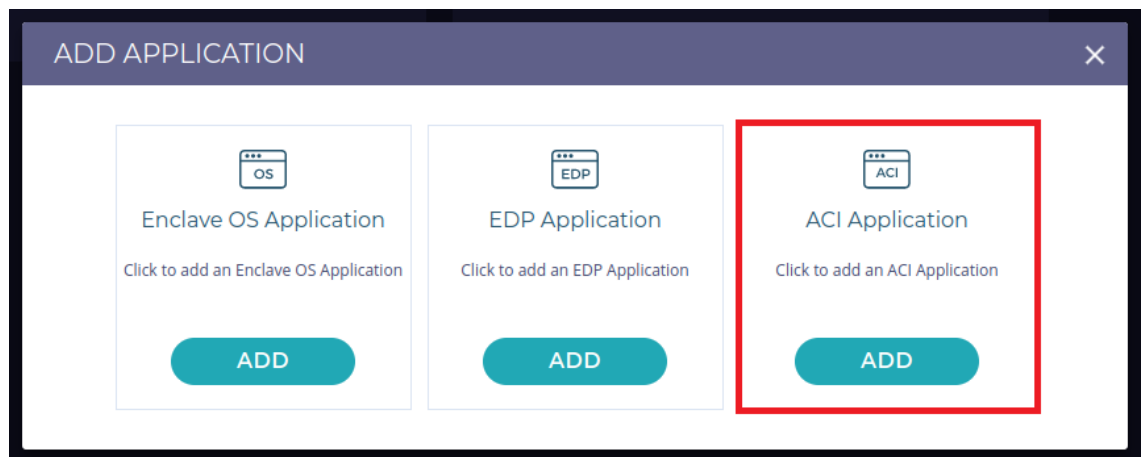


FIGURE 15: ADD ACI APPLICATION

1. In the ACI **Add application** form, fill in the following relevant details:
 - **Application name:** Enter the name of the application.
 - **Description** (optional): Enter the application’s description.
 - **Image name:** Enter the full path of the current application’s docker image. Ensure that the **Image name** does not include any container image tag.

- **Group:** Select the required group name from the drop down menu to associate the application with that Group.
- **Certificate Configuration:** Add any certificate using the **Certificate Configuration** section. A converted application can request a certificate from Fortanix CCM when your application is started. The certificates are signed by the Fortanix CCM Certificate Authority, which issues certificates only to enclaves presenting a valid attestation.
 - **Domain:** Enter the allowed domain for the application. This is the domain that appears in the TLS certificate issued by the Fortanix Confidential Computing Manager.
 - **Type:** Enter the type of the certificate to obtain for the application.

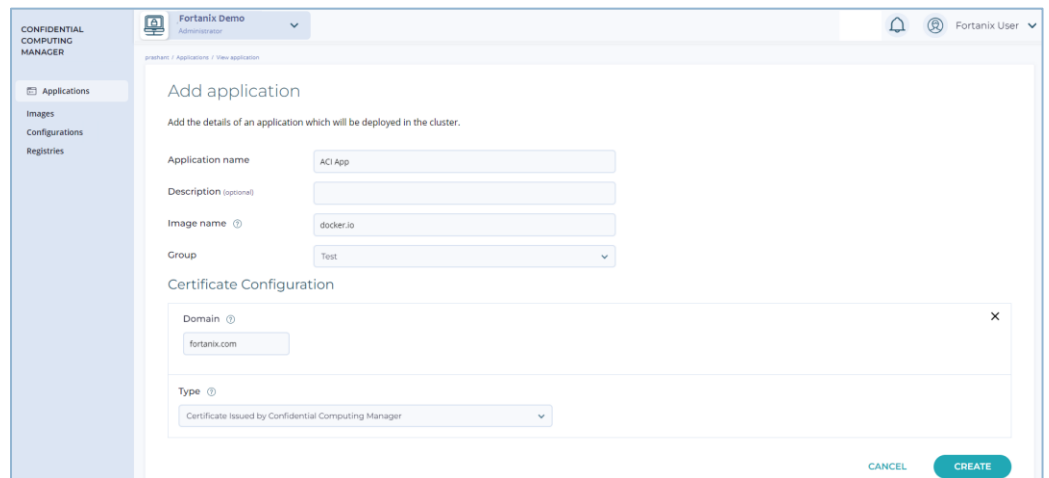


FIGURE 16: ADD ACI APPLICATION DETAILS

2. After you configure the certificate, click the **CREATE** button to configure the application. Refer to [User's Guide: Create an Image](#) to create an image for the ACI application.

4.3.1 EDIT AN ACI APPLICATION

You can edit an ACI application after you add it to your list.

1. Sign in to Fortanix Confidential Computing Manager, and then navigate to the **APPLICATION** tab in the Fortanix Confidential Computing Manager UI.
2. Click the name of the application that you want to edit. A new screen opens where you can review and edit the configuration including certificates.

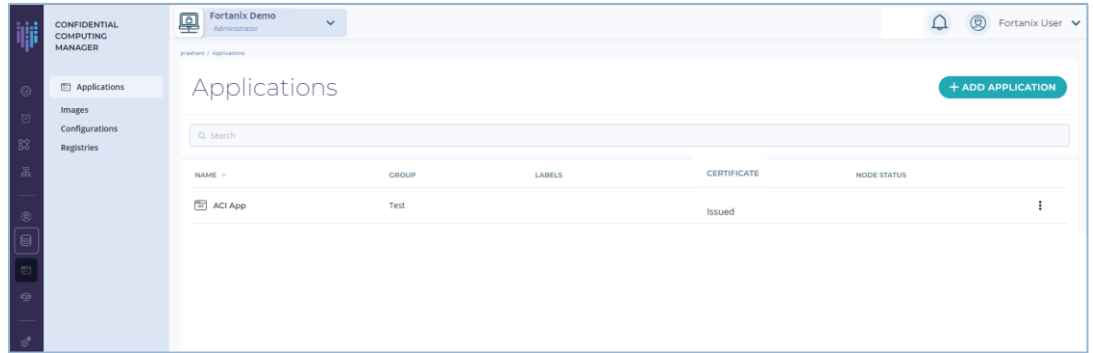


FIGURE 17: VIEW ACI APPLICATION

3. Click **EDIT APPLICATION**.

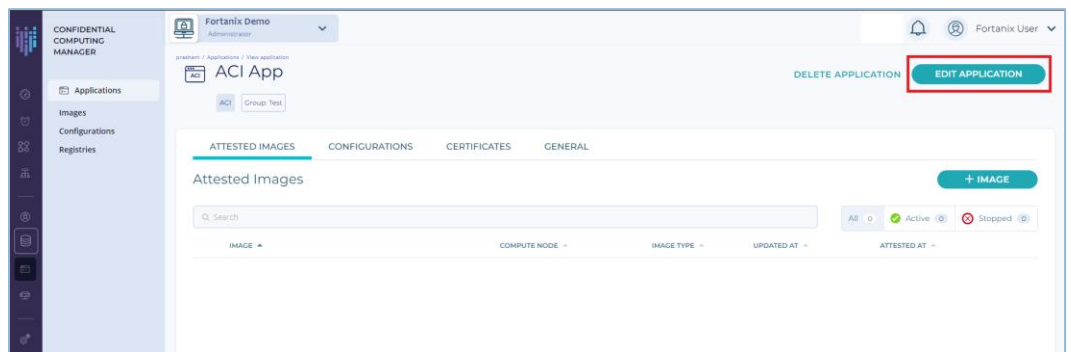


FIGURE 18: EDIT ACI APPLICATION

4. Update the configuration that you want to make.
5. Click the **UPDATE** button.

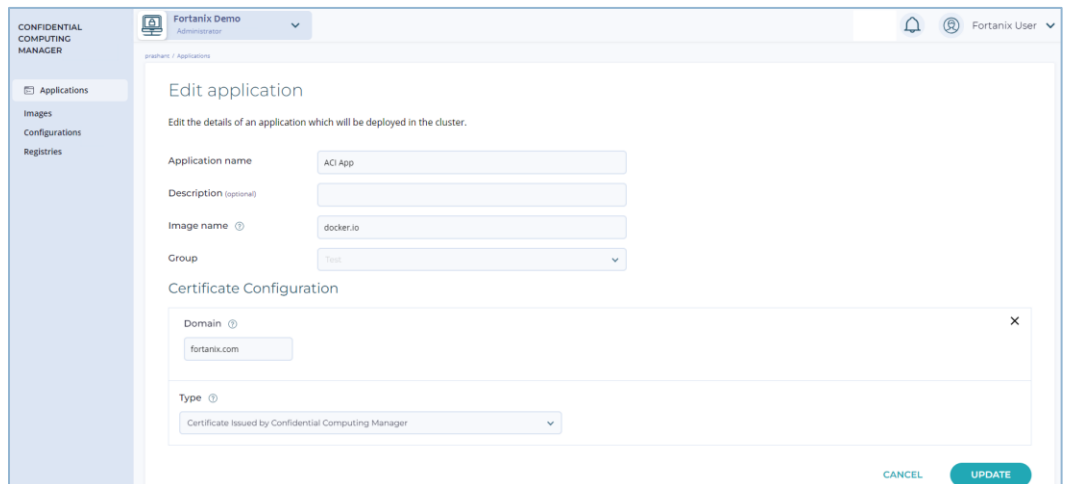


FIGURE 19: UPDATE APPLICATION

4.4 SETTING ENVIRONMENT VARIABLES FOR YOUR APPLICATION

Many applications can be configured by using environment variables such as a container image, a Kubernetes pod specification, or a container entrypoint script. The

`{site.data.keyword.datashield_short}` conversion process transfers any environment variables that are specified by the input container image to a configuration file in the output container, where they are covered by the enclave signature. This freezes the values of the environment variables at conversion time. If variables are supplied after the conversion takes place, they are not seen by the application. Since the variables are not seen, your application is not protected from any maliciously set environment variables at runtime.

By default, the only environment variable passed to the binaries in library OSes is `PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin`. If the host environment variables specifies a `HOSTNAME` then it is also included in the list of default environment variables.

Syntax 1: `loader.env.[ENVIRON]=[VALUE]`

This syntax specifies the environment variable value that is customized for the enclaves. This syntax can be used multiple times to specify more than one environment variable.

The list of environment variables passed to the binaries in enclaves will include a merged list of default environment variables and environment variables specified with this syntax. If there are any conflicting variables, the default environment variable will be overwritten.

Syntax 2: `loader.env.allow_all_env.all = 1`

This syntax passes all the host environment variables to the binaries in the enclaves.

The list of environment variables passed to the binaries in enclaves will include a merged list of host environment variables and variables specified with syntax 1. If there are any conflicting variables, the host environment variables will be overwritten with the value specified by syntax 1. For example, if the manifest specifies `loader.env.X = Z` and the host specifies `X=Y` then the value of `X=Z`.

Syntax 3: `loader.env.allow_some_env.[ENVIRON] = 1`

This syntax specifies the environment variable that will be passed from the host environment variable to the binaries in the enclaves. This syntax can be used multiple times to specify more than one environment variable.

The list of environment variables passed to the binaries in enclaves will include a merged list of a subset of host environment variables as specified by Syntax 3 and variables specified with Syntax 1. If there are any conflicting variables, the host environment variables will be overwritten with the value specified by Syntax 1. For example, if the manifest specifies `loader.env.X = Z` and the host specifies `X=Y` then the value of `X=Z`.

Note that Syntax 2 overrides Syntax 3, so it is recommended to use one or the other of these, not both, in the manifest file.

5.0 DOCUMENT INFORMATION

5.1 DOCUMENT LOCATION

The latest published version of this document is located at the URL:

<https://support.fortanix.com/hc/en-us/articles/360043527431-User-s-Guide-Add-and-Edit-an-Application>

5.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com

© 2016 – 2024 Fortanix, Inc. All Rights Reserved.

Fortanix® and the Fortanix logo are registered trademarks or trade names of Fortanix, Inc.

All other trademarks are the property of their respective owners.

NOTICE: This document was produced by Fortanix, Inc. (Fortanix) and contains information which is proprietary and confidential to Fortanix. The document contains information that may be protected by patents, copyrights, and/or other IP laws. If you are not the intended recipient of this material, please destroy this document and inform info@fortanix.com immediately.