



zenon
by COPA-DATA

Manuel de zenon Docker

v.8.20



© 2020 Ing. Punzenberger COPA-DATA GmbH.

Tous droits réservés.

La distribution et/ou reproduction de ce document ou partie de ce document, sous n'importe quelle forme, n'est autorisée qu'avec la permission écrite de la société COPA-DATA. Les données techniques ne sont utilisées que pour décrire le produit et ne sont pas des propriétés garanties au sens légal. Document sujet aux changements, techniques ou autres.

Contenu

1	Welcome to COPA-DATA help	4
2	Docker	4
3	Notions fondamentales	5
4	Terminologie	5
5	Docker for Windows.....	6
5.1	Installation et configuration	7
5.2	Obtenir Docker Image pour le Runtime zenon depuis Docker Hub	8
5.3	Créer un conteneur avec le Runtime zenon	9
5.4	Configurer le Runtime zenon pour Docker.....	10
5.4.1	Runtime Docker avec résolution du nom.....	13
5.4.2	Ports standard zenon	15
5.5	Une sélection de commandes pour Docker	15

1 Welcome to COPA-DATA help

TUTORIELS VIDÉO DE ZENON.

Des exemples concrets de configurations de projets dans zenon sont disponibles sur notre chaîne YouTube (https://www.copadata.com/tutorial_menu). Les tutoriels sont regroupés par sujet et proposent un aperçu de l'utilisation des différents modules de zenon. Les tutoriels sont disponibles en anglais.

AIDE GÉNÉRALE

Si vous ne trouvez pas certaines informations dans ce chapitre de l'aide ou si vous souhaitez nous suggérer d'intégrer un complément d'information, veuillez nous contacter par e-mail : documentation@copadata.com.

ASSISTANCE PROJET

Vous pouvez obtenir de l'aide pour tout projet en contactant par e-mail notre service d'assistance : support@copadata.com

LICENCES ET MODULES

Si vous vous rendez compte que vous avez besoin de licences ou de modules supplémentaires, veuillez contacter l'équipe commerciale par e-mail : E-mail sales@copadata.com.

2 Docker

Docker est le nom de marque d'un logiciel libre pour l'isolation des applications utilisant la virtualisation des conteneurs. COPA-DATA permet au Runtime zenon d'être utilisé dans un conteneur Docker. Pour ce faire, vous avez besoin de Docker for Windows.

3 Notions fondamentales

Docker simplifie le déploiement des applications. Contrairement à une machine virtuelle, un conteneur Docker ne dispose pas de son propre système d'exploitation. Au lieu de cela, il utilise directement le système d'exploitation de l'hôte.

Toutes les dépendances d'une application sont encapsulées dans une **Image** Docker. Celle-ci contient tous les paquets nécessaires et peut être facilement transportée et installée sous forme de fichier. Les conteneurs assurent l'isolement et la gestion des ressources utilisées sur un ordinateur. Ils sont exécutés localement sur un ordinateur. Les conteneurs que vous créez restent stockés localement après la sortie de l'application et ne sont pas retirés. Si vous souhaitez les supprimer par la suite, les conteneurs peuvent déjà être définis avec un paramètre de suppression automatique lorsque vous appelez le conteneur.

Docker peut être utilisé avec Linux et Windows. Docker for Windows est nécessaire pour le Runtime zenon.

DOCKER ENGINE

Docker Engine donne accès au noyau du système d'exploitation hôte et est capable de créer, de démarrer et d'arrêter des conteneurs. Étant donné que Docker utilise le noyau du système d'exploitation hôte, une image Docker est beaucoup plus petite qu'une machine virtuelle. Les conteneurs Docker peuvent fonctionner sur n'importe quel ordinateur sur lequel Docker Engine a été installé.

4 Terminologie

Termes utilisés dans ce manuel :

Terme	Description
Container :	Instance active d'une image. Le Container est donc actuellement en fonctionnement et occupé. Le conteneur sera fermé automatiquement lorsqu'il n'exécutera plus de programme ou lorsqu'il aura terminé sa tâche.
Container-ID :	ID unique d'un conteneur. Une partie de l'ID est suffisante pour adresser un conteneur, à condition que cette partie identifie clairement un conteneur spécifique.
Docker Engine :	Donne accès au noyau du système d'exploitation hôte et est capable de créer, de démarrer et d'arrêter des conteneurs.

Terme	Description
	Aussi connu comme Docker Daemon .
Docker Hub :	Service en ligne qui contient un Registry pour les images Docker et les Repositories .
Image :	Une image Docker est un fichier, composé de plusieurs couches, qui est utilisé pour exécuter du code dans un conteneur Docker. Une Image est constituée de plusieurs couches qui sont en lecture seule et ne peuvent donc pas être modifiées. Une image est portable, peut être enregistrée dans des Repositories et peut être partagée avec d'autres utilisateurs. Il est toujours possible de démarrer plusieurs conteneurs depuis une seule Image .
Registry :	Utilisé pour l'administration des dépôts. Par exemple, Docker Hub . Le Registry est divisé en une partie publique et une partie privée. Dans la partie publique, chaque utilisateur peut télécharger des images qu'il a créées et les mettre ainsi à la disposition des autres utilisateurs. En attendant, il existe également des images officielles, par exemple, des distributeurs Linux. Dans la partie privée, les utilisateurs peuvent télécharger leurs images Docker et ainsi les distribuer facilement, par exemple, en interne, sans les rendre ainsi accessibles au public.
Repository :	Ensemble d'images du même nom avec des tags différents, le plus souvent des versions.
Tag :	Utilisé pour identifier les différentes versions d'une image. Cette définition ne s'applique qu'en lien avec Docker. Elle n'est pas utilisée pour les termes en rapport avec les variables ou les paramètres de Batch Control.

5 Docker for Windows

Docker for Windows permet au Runtime zenon d'être utilisé dans une image Docker.

5.1 Installation et configuration

Pour exécuter zenon dans un Docker Container, vous avez besoin en même temps de zenon, de Hyper-V et de Docker.

COPA-DATA ne peut fournir aucune aide pour l'installation de Hyper-V et de Docker.

Vous pouvez trouver des informations sur Docker for Windows, par exemple, aux adresses suivantes :

- ▶ Informations générales : <https://docs.docker.com/docker-for-windows/>
(<https://docs.docker.com/docker-for-windows/>)
- ▶ Réseau : <https://docs.docker.com/network/> (<http://www.>)
- ▶ Fonctionnalités réseau : <https://docs.docker.com/docker-for-windows/networking/>
(<https://docs.docker.com/docker-for-windows/networking/>)

CONDITIONS REQUISES

Pour utiliser Docker avec zenon, vous aurez besoin des éléments suivants :

- ▶ système d'exploitation approprié
 - ▶ Windows 10 Pro ou supérieur
 - ▶ Windows Server 2016 ou supérieur
- ▶ environnement de virtualisation active
- ▶ une licence valide

Remarque : zenon ne peut pas être exécuté en mode démo dans un conteneur Docker.

Pour la configuration, vous aurez besoin des éléments suivants :

- ▶ Docker Desktop
- ▶ installation zenon disponible
- ▶ PowerShell

INSTALLER ET CONFIGURER DOCKER

Pour installer Docker for Windows :

1. Chargez Docker for Windows depuis le site Web
<https://docs.docker.com/docker-for-windows/install/> (<http://www.>)
Vous avez besoin de la version 2.0.3 ou plus.
2. Installez le paquet.
Attention :
 - ▶ L'option **Use Windows containers instead of Linux containers Windows-Container** doit être activée sur la page de configuration.

- ▶ Une fois l'installation terminée, vous serez déconnecté.
 - ▶ Une fois que vous vous reconnectez, vérifiez si Hyper-V est activé.
Si ce n'est pas le cas, activez Hyper-V.
3. Cliquez avec le bouton droit sur l'icône docker dans la barre système de la barre des tâches.
 4. Dans le menu contextuel, sélectionnez **Switch to Windows container...**
Si cette entrée n'est pas affichée, c'est que Docker Desktop est déjà configuré correctement.
 5. Vérifiez si Docker est correctement installé.
Pour ce faire :
 - ▶ Ouvrez un Powershell dans Windows.
 - ▶ Entrée **docker --version**
Les informations relatives à la version sont affichées.

COMMUTATEUR EXTERNE HYPER-V

Un commutateur NAT est utilisé par défaut pour les conteneurs Docker. Cependant, vous adresserez votre Runtime zenon à partir d'autres clients via le nom d'hôte du conteneur Docker.

Pour valider votre conteneur Docker pour le serveur DHCP et DNS :

1. Créez un commutateur externe dans Hyper-V dans le **Virtual Switch Manager**.
2. Restart Docker Desktop.
De cette façon, le commutateur virtuel est reconnu.
3. Vérifiez si le nouveau commutateur est disponible.
Pour ce faire, saisissez **docker network ls** dans PowerShell

Vous trouverez de plus amples informations sur les réseaux zenon avec Docker dans la section **Runtime Docker avec résolution de nom** (à la page 13).

5.2 Obtenir Docker Image pour le Runtime zenon depuis Docker Hub

Vous trouverez l'image pour le Runtime zenon sur <https://hub.docker.com/u/copadata> (<https://hub.docker.com/u/copadata>).

Pour charger l'image depuis Docker Hub :

1. Ouvrez PowerShell.
2. Sélectionnez la version de Runtime souhaitée.
3. Téléchargez l'image avec la **Docker Pull Command** affichée.

4. Vous recevrez des informations sur l'empreinte digitale et l'état du téléchargement.

5.3 Créer un conteneur avec le Runtime zenon

Créez maintenant un conteneur avec un Runtime zenon. Le conteneur contient une référence aux fichiers zenon. Les fichiers journaux, les fichiers d'exécution et les licences y sont stockés.

Pour créer le conteneur :

1. Sur votre hôte, créez un dossier dans lequel sont stockés les fichiers Runtime de votre conteneur Runtime. Les informations sur les paramètres et les projets y sont stockées. Ce dossier est appelé **data path** dans la documentation. Vous pouvez trouver la structure dans le fichier **zenonDataTemplate.zip** sur le support d'installation.
2. Copiez les fichiers Runtimes dans le sous-dossier **data-path\projects**.
Si ce dossier est vide, les fichiers seront créés dans le conteneur la première fois que le Runtime est lancé.
3. Configurez l'installation zenon. Pour ce faire, éditez les fichiers dans **data path**.
Vous trouverez des informations détaillées à ce sujet au chapitre **zenon Configurer le Runtime pour Docker** (à la page 10).
4. Modifiez les valeurs dans **zenon6.ini (data path\system\zenon6.ini)** :
 - ▶ **VBF30**: chemin vers le projet Runtime.
Par exemple : **C:\zenondata\projects[workspace]**
 - ▶ **DEFANWENDUNG30**: Nom du projet Runtime.
5. Modifiez **settings.xml (data-path\settings.xml)**.
Vous pouvez configurer ici le serveur de licence, le nom DNS primaire et les processus à lancer.
6. Ouvrez PowerShell.
7. Saisissez la commande suivante : **docker run -d -v [data-path]:C:\zenondata --name [Nom du conteneur] copadata/zenon-runtime820-windows**
 - ▶ **data-path**: chemin d'accès au dossier qui doit contenir les données sauvegardées. Le chemin que vous avez créé lors de la première étape.
 - ▶ **Nom du conteneur**: nom du conteneur nouvellement créé, par exemple : zenon

Exemple : **docker run -d -v C:\zenondata:C:\zenondata --name zenon copadata/zenon-runtime820-windows**

De cette façon, les fichiers sont créés dans le dossier **[data-path]**.

- ▶ Arrêtez le conteneur.
Pour ce faire, saisissez : **docker stop [Nom du conteneur]**
Nom du conteneur : Nom du conteneur qui a été créé auparavant avec **docker run**.

5.4 Configurer le Runtime zenon pour Docker

CONDITIONS REQUISES

Pour pouvoir utiliser zenon dans un conteneur Docker, les services requis doivent être disponibles. Ceux-ci sont automatiquement activés lors de l'installation.

Il est ainsi possible :

- ▶ de transférer des fichiers Runtime par transport à distance.
- ▶ de lire des fichiers Runtime par transport à distance.
- ▶ de démarrer, recharger et fermer automatiquement le Runtime.

DOSSIERS ET FICHIERS

Pour commencer, tous les sous-dossiers et fichiers nécessaires doivent être disponibles via le **chemin d'accès aux données** :

- ▶ **System\K5LICENSE.INI** : licence de zenon Logic.
- ▶ **System\License.ini** : licence de zenon.
- ▶ **System\zenon6.ini** : paramètres de zenon
- ▶ **settings.xml** : Paramètres pour le serveur de licence, le comportement de démarrage du conteneur et le Primary DNS-Suffix.
- ▶ **Projects** : collecte les dossiers de projets.
- ▶ **LOG** : collecte les fichiers journaux.

Le serveur de licence configuré est vérifié au démarrage.

Si une licence réseau valide est disponible, le Runtime est sous licence. Si aucune licence valide n'est disponible, le Runtime ne peut pas être lancé.

ENREGISTREMENT (FICHER JOURNAL)

Si une fonctionnalité utilisée dans le Runtime n'est pas prise en charge par Docker, une entrée est insérée dans le fichier journal.

Emplacements :

- ▶ zenon fichiers log : ... \ *[chemin d'accès aux données]* \ LOG
Peut être visualisé avec Diagnosis Viewer.
- ▶ Fichiers log Docker : *Windows Logs\Application ...*
Peut être visualisé avec Windows Event Viewer.

CONFIGURATION

Pour configurer zenon afin de l'utiliser dans une image Docker, les fichiers **settingx.xml**, **license.ini** et **zenon6.ini** doivent être modifiés à l'aide d'un éditeur de texte.

SETTINGS.XML

Dans le fichier settings.xml, vous pouvez configurer :

- ▶ Le serveur de licence CodeMeter : Cette entrée est requise par **license.ini**.
- ▶ Suffixe DNS primaire : Pour le transfert de réseau.

Pour configurer settings.xml :

1. Suivez le chemin ...\ [chemin d'accès aux données]\.
2. Ouvrez le fichier **settings.xml**.
3. Accédez à la section **<CodeMeterLicenseServer>**.
 - ▶ Saisissez le nom de l'ordinateur sur lequel la licence est stockée entre le Tags **<Address>** et **</Address>**.
4. Répétez cette étape si nécessaire pour une entrée supplémentaire sur le serveur.
5. Passez à la section **<Network>**.
 - ▶ Entrez le suffixe DNS primaire de votre réseau entre les Tags **<PrimaryDnsSuffix>** et **<PrimaryDnsSuffix>**.
C'est la partie de votre entrée DNS sans le nom d'hôte.
Si cette entrée reste vide, les paramètres par défaut du conteneur seront utilisés.
6. Si nécessaire, vous pouvez configurer le démarrage automatique du Runtime dans le conteneur.
Pour ce faire, entrez les informations de service et de programme suivantes entre les tags **<Autostart>** et **</Autostart>**.
 - ▶ **<Program>** : initie la section pour le démarrage du service.
 - ▶ **<Active></Active>** : active le démarrage automatique du service.
Actif : *true*
Inactif : *false*
Par exemple : **<Active>true</Active>**.
 - ▶ **<Path></Path>** : chemin d'accès au service.
Par exemple, **<Path>C:\zenonSetup\64\zenSysSrv.exe</Path>**.
 - ▶ **</Program>** : ferme la section.
 - ▶ **<Program>** : initie la section pour le démarrage du programme.

- ▶ **<Active></Active>** : active le démarrage automatique pour le programme de manière similaire à celui du service.
Par exemple : **<Active>true</Active>**.
- ▶ **<Path></Path>** : chemin d'accès au programme.
Par exemple, **<Path>C:\zenonSetup\64\zenrt32.exe</Path>**.
- ▶ **</Program>** : ferme la section.

7. Enregistrez le fichier.

LICENSE.INI SAISIR LICENCE

Afin de pouvoir utiliser zenon dans un conteneur Docker, vous devez fournir une licence valide et définir l'endroit où la licence est enregistrée.

Sources d'informations pour :

- ▶ Licence : certificat de licence ou gestionnaire de licences
- ▶ Emplacement de stockage de la licence : Codemeter Web Admin

Pour vous procurer une licence pour zenon pour Docker :

1. Suivez le chemin ...\
[chemin d'accès aux données]\system.
2. Ouvrez le fichier **License.ini**.
3. Saisissez les informations relatives à la licence en suivant votre certificat de licence :
 - ▶ **SERIAL0** = : Numéro de série de votre licence.
Cette entrée est obligatoire.
 - ▶ **SERIAL0_DESC** = : description facultative de l'utilisation de la licence.
 - ▶ **SERIAL0_LOCATION** = : Adresse du serveur de licence.
Cette entrée est obligatoire.
4. Enregistrez le fichier.
5. Arrêtez et redémarrez le conteneur Docker.
Saisissez : **docker stop [container name]** et **docker start [container name]**

Un zenon sous licence est à votre disposition dans le conteneur Docker.

ZENON6.INI

Pour spécifier le projet Runtime :

1. Suivez le chemin ...\
[chemin d'accès aux données]\system.
2. Lancez le **zenon6.ini**.
3. Éditez les entrées suivantes :

[PATH]

VBF30= Entrez ici le chemin d'accès à l'espace de travail.

Exemple : **C:\zenondata\projects\Workspace1**

[DEFAULT]

DEFANWENDUNG30= Entrez le nom du projet ici.

5.4.1 Runtime Docker avec résolution du nom

Pour que le serveur de Runtime zenon fonctionne et soit accessible, il faut un conteneur Docker auquel on peut accéder de l'extérieur via le nom d'hôte du conteneur. Si l'accès passe à travers des domaines, le nom d'hôte complet (Fully Qualified Domain Name, FQDN) doit être utilisé.

Remarque : Conformément aux conventions de Windows, les noms d'hôte ne peuvent contenir plus de 15 caractères.

CONDITIONS REQUISES

Les éléments suivants sont nécessaires :

- ▶ une image Docker du Runtime zenon préalablement chargée.
- ▶ un Runtime zenon qui a déjà été configuré (à la page 10) pour Docker. Avant tout, **settings.xml** et **license.ini** doivent être configurés.

CONFIGURER UN COMMUTATEUR EXTERNE VIRTUEL DANS HYPER-V

Vous avez besoin d'un commutateur de réseau externe dans votre serveur Hyper-V. Vérifiez-le dans le paramètre **Virtual Switch Manager** sur le serveur Hyper-V. Si aucun commutateur externe n'est disponible, vous devez en créer un. Ce commutateur externe est utilisé par le conteneur Docker.

DÉFINIR LE SUFFIXE DNS PRIMAIRE DANS LE CONTENEUR DOCKER

Si vous souhaitez utiliser un nom de serveur avec un suffixe DNS, vous devez communiquer le **Primary DNS Suffix** au conteneur Docker. Par exemple : **zenRT001.domain.internal**. Si le **Primary DNS Suffix** n'est pas défini, le Runtime zenon ne reconnaît pas qu'il joue le rôle de serveur. C'est également le cas si le conteneur Docker peut être interrogé de l'extérieur.

Vous pouvez configurer le suffixe dans **Settings.xml** (à la page 10) dans Tag **<Settings> <Network> <PrimaryDnsSuffix>**.

CRÉER UN CONTENEUR AVEC UN COMMUTATEUR EXTERNE

Vous pouvez créer un conteneur avec un commutateur externe en utilisant la commande suivante :

```
docker run -d -v [data path]c:\zenondata --network [name of external switch] --hostname [unique host name] --name [container name] copadata/zenon-runtime820-windows
```

EXEMPLE

Un conteneur remplissant les conditions suivantes doit être créé :

- ▶ utilise le commutateur virtuel Hyper-V **zrtExtSwitch**
- ▶ définit le nom d'hôte sur **zenRT001**
- ▶ peut gérer le conteneur sous le nom **zenon**

Commande: **PS> docker run -d -v C:\zenondata:C:\zenondata --network zrtExtSwitch --hostname zenRT001 --name zenon copadata/zenon-runtime820-windows**

CONFIGURER LE SERVEUR ZENON

Pour configurer le serveur zenon :

- ▶ utilisez le nom d'hôte **zenRT001.domain.internal** précédemment configuré comme nom de serveur dans votre projet de réseau zenon.
- ▶ Vous pouvez utiliser le même nom d'hôte pour le transport à distance zenon.
- ▶ Après le transfert des fichiers Runtime et le lancement du Runtime zenon dans le conteneur Docker, les clients zenon peuvent accéder au serveur via **zenRT001.domain.internal**.

DÉPANNAGE

En cas d'erreurs :

- ▶ Veillez à ce que le Runtime zenon se reconnaisse comme le serveur.
Pour ce faire, consultez les fichiers journaux.
- ▶ Vérifiez la connexion réseau entre le client et le serveur.

TRANSPORT À DISTANCE : LE REDÉMARRAGE DU CONTENEUR EMPÊCHE LA CONNEXION

Lorsqu'un conteneur Docker est redémarré, il reçoit une nouvelle adresse IP. Cependant, le cache DNS pour le nom d'hôte du conteneur n'est pas mis à jour en même temps. Les connexions avec le conteneur par nom d'hôte ne sont donc plus possibles.

Solutions :

1. Dans le cadre du même domaine : Exécutez la ligne de commande **ipconfig /flushdns**.

2. En règle générale : Attendez que le cache DNS ait été mis à jour après le timeout défini.

5.4.2 Ports standard zenon

Ports utilisés par zenon par défaut :

Application	Port standard
Network Service	1100
Transport Service	1101
WEB Service Classic	1102
DB Service	1103
SQL Browser Service, (pour le développement distribué dans Editor)	1434
zenAdminSrv.exe	50777
zenLicTransfer (Service de transfert de licence)	50784
Logging Service	50780
SNMP Trap Service	50782
WEB Service Tunneling	8080

5.5 Une sélection de commandes pour Docker

Vous avez besoin de plusieurs commandes pour la configuration de zenon avec Docker. Vous les saisissez via Windows PowerShell. Le tableau suivant contient les commandes les plus fréquemment utilisées.

Vous trouverez également des informations à ce sujet à l'adresse suivante : <https://docs.docker.com/engine/reference/commandline/docker/> (<http://www.>)

Commande	Description
<code>docker --version</code>	Affiche les informations sur la version de Docker.
<code>docker network ls</code>	Liste tous les réseaux que engine daemon reconnaît. Cela comprend les réseaux qui s'étendent sur plusieurs hôtes dans un

Commande	Description
	groupe.
<code>docker images</code>	Liste toutes les Images chargées.
<code>docker run -d -v [data path]:c:\zenondata --name [container name] [image name]</code>	Crée un conteneur à partir d'une Image chargée.
<code>docker run -d -v [data path]:c:\zenondata --network [name of switch] --hostname [host name] --name [container name] [image name]</code>	Crée un conteneur à partir d'une Image chargée avec transfert des ports pour le réseau zenon.
<code>docker ps -a</code>	Liste tous les conteneurs, y compris ceux qui ont été arrêtés.
<code>docker start [container name]</code>	Démarre un conteneur arrêté.
<code>docker stop [container name]</code>	Arrête un conteneur démarré. Conseil : Pour arrêter tous les conteneurs, utilisez la commande <code>\$(docker ps -a -q) as [container name]</code> .
<code>docker inspect [container name]</code>	Affiche les détails du conteneur, y compris le nom d'hôte et la configuration du réseau.
<code>docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' [container name]</code>	Indique l'adresse IP statique du conteneur.
<code>docker rm [container name]</code>	Supprime un conteneur. Conseil : Pour supprimer tous les conteneurs, utilisez la commande <code>\$(docker ps -a -q) as [container name]</code> .
<code>docker exec -it [container name] powershell</code>	Permet d'exécuter les commandes PowerShell dans le conteneur Docker.
<code>docker system prune -a</code>	Cette commande supprime : <ul style="list-style-type: none"> ▶ tous les conteneurs arrêtés ▶ tous les réseaux qui ne sont pas utilisés par au moins un conteneur ▶ toutes les images qui ne sont pas liées à au moins un conteneur ▶ tous les caches de versions

