Docker 環境での ASA コンテナの展開

任意のクラウドプラットフォームで実行されるオープンソースの Docker 環境で ASA コンテナ（ASAc）を展開できます。

- 概要 （1 ページ）
- Docker 環境で ASA コンテナを展開するためのガイドラインと制限事項 （1 ページ）
- Docker 環境で ASA コンテナを展開するためのライセンス （2 ページ）
- Docker 環境で ASA コンテナを展開するためのソリューションのコンポーネント （2 ページ）
- Docker 環境で ASA コンテナを展開するためのサンプルトポロジ （3 ページ）
- Docker 環境で ASA コンテナを展開するための前提条件 （4 ページ）
- Docker 環境での ASA コンテナの展開 （4 ページ）
- Docker 環境での ASA コンテナ展開の検証 （6 ページ）
- Docker 環境での ASA コンテナ展開ログへのアクセス （6 ページ）
- Docker 環境での ASA コンテナへのアクセス （7 ページ）

概要

コンテナは、コンピューティング環境でアプリケーションが正常に実行されるようにするためのコードと関連する要件（システムライブラリ、システムツール、デフォルト設定、ランタイムなど）をバンドルしたソフトウェアパッケージです。Cisco Secure Firewall ASA バージョン 9.22 以降では、オープンソースの Docker 環境で ASA コンテナ（ASAc）を展開できます。

Docker環境でASAコンテナを展開するためのガイドラインと制限事項

- ASA コンテナソリューションは、オープンソースの Kubernetes および Docker 環境でのみ検証されます。
- EKS、GKE、AKS、OpenShift などの他の Kubernetes フレームワークは、まだ検証されていません。

- アップグレードは、新しいコンテナイメージを使用してローリングアップグレードとして実行されます。
- ASA コンテナの再起動はサポートされていません。
- 次の機能は検証されていません。
  - クラスタ
  - トランスペアレント モード
  - サブインターフェイス

Docker環境でASAコンテナを展開するためのライセンス

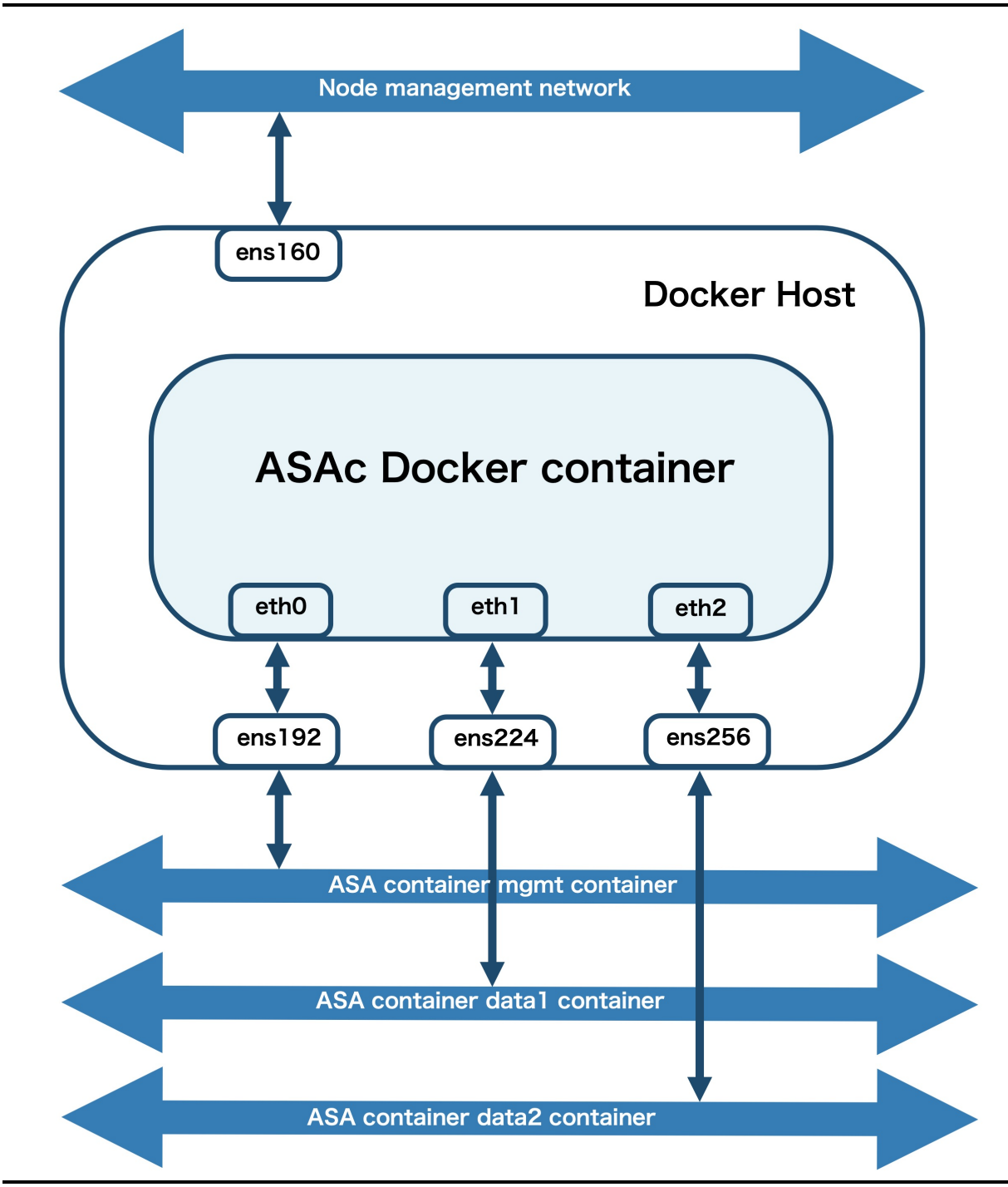
次のいずれかのライセンスを使用すると、Docker で ASA コンテナを展開できます。

- ASAc5：1 vCPU、2 GB RAM、および 100 Mbps のレート制限
- ASAc10：1 vCPU、2 GB RAM、および 1 Gbps のレート制限

Docker 環境で ASA コンテナを展開するためのソリューションのコンポーネント

- オペレーティング システム
  - Docker ホスト上の Ubuntu 20.04.6 LTS
- 設定検証用の Macvlan ネットワーク

Docker環境でASAコンテナを展開するためのサンプルトポロジ



このサンプルトポロジでは、ASA Docker コンテナに 3 つの仮想ネットワーク インターフェイス（eth0、eth1、eth2）があり、インターフェイス ens192、ens224、および ens256 に接続されています。これらのインターフェイスは、ASAc mgmt、data1、および data2 ネットワークにマッピングされます。インターフェイス ens160 は、ノード管理インターフェイスです。

Docker 環境で ASA コンテナを展開するための前提条件

- Ubuntu 20.04.6 LTS が Docker ホストにインストールされていることを確認します。
- ASA コンテナの操作のために、Docker ホストに 3 つの仮想インターフェイスを割り当てます。
- Docker ホストへの SSH アクセスに使用する Docker ホストの管理インターフェイスをセットアップします。
- Docker ホストで Hugepages を有効にします。
- 設定検証用の macvlan ネットワークを使用して Docker バージョン 24.0.5 をセットアップします。

これらの前提条件に記載されている一般的な Docker 操作の詳細については、Docker のドキュメントを参照してください。

Docker 環境での ASA コンテナの展開

Docker 環境で ASA コンテナ（ASAc）を展開するには、次の手順を実行します。

手順

ステップ 1 前提条件に記載されている要件をセットアップします。

ステップ 2 **route -n** コマンドを実行し、ネットワーク インターフェイス構成を確認します。この例では、ens160 はノードの管理インターフェイスです。ノード ens192、ens224、および ens256 は、ASAc インターフェイスにマッピングされます。

（注）
以下に示す出力は、サンプル出力のみです。

Docker 環境での ASA コンテナの展開

```
ubuntu@k8s-worker:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.10.4.1       0.0.0.0         UG    100    0        0 ens160
10.10.4.0       0.0.0.0         255.255.255.224 U     0      0        0 ens160
10.10.4.1       0.0.0.0         255.255.255.255 UH    100    0        0 ens160
10.10.4.32      0.0.0.0         255.255.255.224 U     0      0        0 ens192
10.10.4.64      0.0.0.0         255.255.255.224 U     0      0        0 ens224
10.10.4.96      0.0.0.0         255.255.255.224 U     0      0        0 ens256
10.244.235.192  10.244.235.192  255.255.255.192 UG    0      0        0 vxlan.calico
10.244.254.128  0.0.0.0         255.255.255.192 U     0      0        0 *
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
```

ステップ 3 次に示す **cat** コマンドを実行し、hugepage 設定を確認します。

```
ubuntu@k8s-worker:~$ cat /proc/meminfo | grep -E 'HugePages_Total|HugePages_Free'
HugePages_Total:    2048
HugePages_Free:     2048
```

ステップ 4 ASA コンテナイメージを含む ASA Docker tar バンドルを、software.cisco.com からダウンロードします。

ステップ 5 ホストで Docker tar バンドルをロードします。

```
$ docker load < asac9-22-1-1.tar
$ docker images
   REPOSITORY                                  TAG         IMAGE ID
   dockerhub.cisco.com/asac-dev-docker/asac    9.22.1.1    55f5dbc5f3aa
```

ステップ 6 ASAc GitHub リポジトリの **docker** フォルダからテンプレートなどをダウンロードします。

ステップ 7 **docker network create** コマンドを実行して、Docker ネットワークを作成します。ASAc には、内部と外部のネットワーク用に 1 つの管理インターフェイスと 2 つのデータインターフェイスが必要です。Docker を起動すると、Docker ネットワークがアルファベット順に Docker に接続されます。管理インターフェイスには、Docker に接続される最初のインターフェイスになるように名前を付けることをお勧めします。

```
$ docker network create -d macvlan -o parent=ens192 asac_nw1
$ docker network create -d macvlan -o parent=ens224 asac_nw2
$ docker network create -d macvlan -o parent=ens256 asac_nw3
```

ステップ 8 **docker network ls** コマンドを実行して、ネットワークが正常に作成されたことを確認します。

```
$ docker network ls
NETWORK ID    NAME       DRIVER   SCOPE
06f5320016f8  asac_nw1   macvlan  local
258954fa5611  asac_nw2   macvlan  local
3a3cd7254087  asac_nw3   macvlan  local
```

ステップ 9 **day0-config** ファイルに存在するデフォルトのパラメータ値を確認します。必要に応じて、これらの値を更新することもできます。

ステップ 10 必要に応じて、**start_docker_asac.sh** スクリプトを開き、CPU、メモリ、コンテナ名、およびイメージリポ名の設定値を更新します。

（注）

start_docker_asac.sh スクリプトのパラメータには、デフォルトの設定値が指定されています。必要な場合にのみ変更してください。

ステップ 11 次のコマンドを実行して、Docker 環境で ASAc を起動します。

```
$ ./<script-name> <asac-image-path-and-version> <asac-mgmt-nw> <asac-data1-nw> <asac-data2-nw>
```

```
$ ./start_docker_asac.sh dockerhub.cisco.com/asac-dev-docker/asac:9.22.1.1 asac_nw1 asac_nw2
asac_nw3
   Docker networks are provided..
   Starting ASA Build Container...
   docker create -it --privileged --cap-add=NET_RAW --network asac_nw1 --name asac -e ASAC_CPUS=1
-e ASAC_MEMORY=2048M -v /dev:/dev -v /home/ubuntu/standalone-asac/docker/day0-config:/asacday0-
config/day0-config:Z -v /home/ubuntu/standalone-asac/docker/interface-config:/mnt/disk0/
interface-config/interface-config:Z -e CORE_SIZE_LIMIT=200MB -e COREDUMP_PATH=/mnt/coredump_repo/
-e ASA_DOCKER=1 -e ASAC_STANDALONE_MODE=1 -e ASAC_ROOT_PRIVILEGE=1 --entrypoint /asa/bin/
lina_launcher.sh dockerhub.cisco.com/asac-dev-docker/asac:9.22.1.1

    Mount Points:
    ------------------------------------------------------------------------------------------
    Host                                                    Container
    ----                                                    ---------
    /dev                                                    /dev
    /home/ubuntu/standalone-asac/docker/day0-config        /asac-day0-config/day0-config
    /home/ubuntu/standalone-asac/docker/interface-config
/mnt/disk0/interface-config/interface-config
------------------------------------------------------------------------------------------
docker network connect asac_nw2 asac
docker network connect asac_nw3 asac
docker start asac
```

Docker 環境での ASA コンテナ展開の検証

Docker ホストで実行されているコンテナのリストを確認することで、ASA コンテナが正常に展開されているかどうかを検証します。

```
$ docker ps -a
CONTAINER ID IMAGE                                                      COMMAND
    CREATED       STATUS        PORTS  NAMES
6e5bff4dbcaf dockerhub.cisco.com/asac-dev-docker/asac:9.22.x.x   "/asa/bin/lina_launc…"
 3 minutes ago Up 3 minutes         asac
```

Docker 環境での ASA コンテナ展開ログへのアクセス

何らかの問題が発生した場合は、トラブルシュートのために **docker logs asac** コマンドを実行して Docker ログを確認してください。

```
$ docker logs asac
Skip NVMe Device for ASAc mode
cdrom device /dev/sr0 found
mount: /mnt/cdrom: WARNING: source write-protected, mounted read-only.
```

```
Error: Encrypted file system support not in Linux kernel.
nr_overcommit_hugepages set to 128 for virtual platform
info: ASAc SSHd Directory Created
No interface-config file found at /interface-config, using default shared
file: /mnt/disk0/interface-config/interface-config
No day0-config file found at /day0-config, using default shared file:
/asac-day0-config/day0-config
info: ASAc Day 0 configuration installed.
info: ASAc Primay/backup Key installed
info: Running in vmware virtual environment.
....
INFO: Network Service reload not performed.
INFO: Power-On Self-Test in process.
.....................................
INFO: Power-On Self-Test complete.
INFO: Starting SW-DRBG health test...
INFO: SW-DRBG health test passed.
Creating trustpoint "_SmartCallHome_ServerCA" and installing certificate...
Trustpoint CA certificate accepted.
Creating trustpoint "_SmartCallHome_ServerCA2" and installing
certificate...
Trustpoint CA certificate accepted.
User enable_1 logged in to ciscoasa
Logins over the last 1 days: 1.
Failed logins since the last login: 0.
Type help or '?' for a list of available commands.
ciscoasa>
```

Docker 環境での ASA コンテナへのアクセス

docker attach asac コマンドを実行して ASA コンテナ（ASAc）の CLI にアクセスし、必要な出力を取得します。この例では、ASAc の CLI にアクセスして **show version** コマンドを実行しています。

（注） ASDM を使用して Docker 環境で ASAc にアクセスすることもできます。

```
ciscoasa> enable
Password: *********
ciscoasa# sh version
Cisco Adaptive Security Appliance Software Version 9.22
SSP Operating System Version 82.16(0.216i)
Device Manager Version 7.22
Compiled on Tue 28-Nov-23 14:37 GMT by builders
System image file is "Unknown, monitor mode tftp booted image"
Config file at boot was "startup-config"
ciscoasa up 9 mins 50 secs
Start-up time 36 secs
Hardware: ASAc, 2048 MB RAM, CPU Xeon E5 series 2100 MHz, 1 CPU (1
core)
BIOS Flash Firmware Hub @ 0x1, 0KB
0: Ext: Management0/0 : address is 0242.ac12.0002, irq 0
1: Ext: GigabitEthernet0/0 : address is 0242.ac13.0002, irq 0
2: Ext: GigabitEthernet0/1 : address is 0242.ac14.0002, irq 0
3: Int: Internal-Data0/0 : address is 0000.0100.0001, irq 0
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。