

第4回 Cisco NSO サクセスコミュニティ ハンズオン 最新版 NSO コンテナを動かしてみよう！



Customer Success Specialist,
Hirotugu Takahashi

July 25, 2025

Disclaimer

This document is Cisco Confidential information provided for your internal business use in connection with the Cisco Services purchased by you or your authorized reseller on your behalf. This document contains guidance based on Cisco's recommended practices.

You remain responsible for determining whether to employ this guidance, whether it fits your network design, business needs, and whether the guidance complies with laws, including any regulatory, security, or privacy requirements applicable to your business.

免責

この文書は、お客様またはお客様の代理人である認定リセラーが購入したCiscoサービスに関連して、お客様が社内業務において使用することを目的としてCiscoが提供するCiscoの機密情報です。この文書にはCiscoが推奨するプラクティスに基づく手引きが記載されています。

お客様は、この手引きを使用するか否かやお客様のネットワーク設計および業務上のニーズにこの手引きが適合しているか否か、さらにはこの手引きが法律(お客様の業務に適用される規制上の要件、セキュリティ上の要件およびプライバシーに関する要件を含みます)に準拠しているか否かを判断する責任を引き続き負います。

dCloudへのアクセス

dCloud へのアクセス

- dCloud はシスコが提供するクラウドベースのデモ環境です
- 今回のハンズオンは、参加者それぞれに独立した環境を用意しています
- ブラウザだけで作業ができるようになっています

NSO 101 hands-on lab with 2 XRd routers (NSO 6.5-final-python3.12-fix-sym)

Start: 22-Jul-2025 07:44 End: 29-Jul-2025 09:20 Session ID: 611472 Virtual Center: 3 Event Name: NSO Success Community

Info End Save Reset Edit Shared View

dCloud へのアクセス

- まずは環境が正常に動作している確認します
- ブラウザを開き下記 URL にアクセスします
- <https://dcloud2-sng.cisco.com/dashboard/sessions>
- すると、ハンズオンのセッションが共有されていると思いますので View ボタンをクリックします

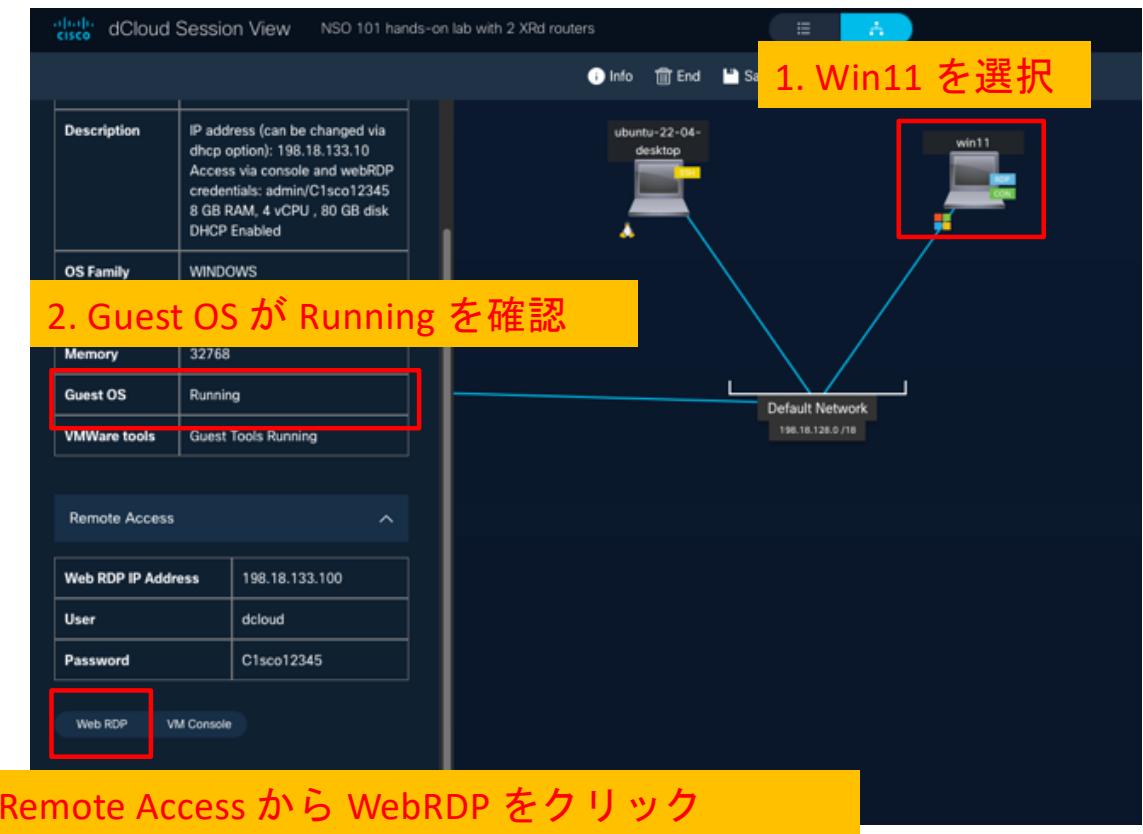
NSO 101 hands-on lab with 2 XRd routers (NSO 6.5-final-python3.12-fix-sym)

Start: 22-Jul-2025 07:44 End: 29-Jul-2025 09:20 Session ID: 611472 Virtual Center: 3 Event Name: NSO Success Community

Info End Save Reset Edit Shared View

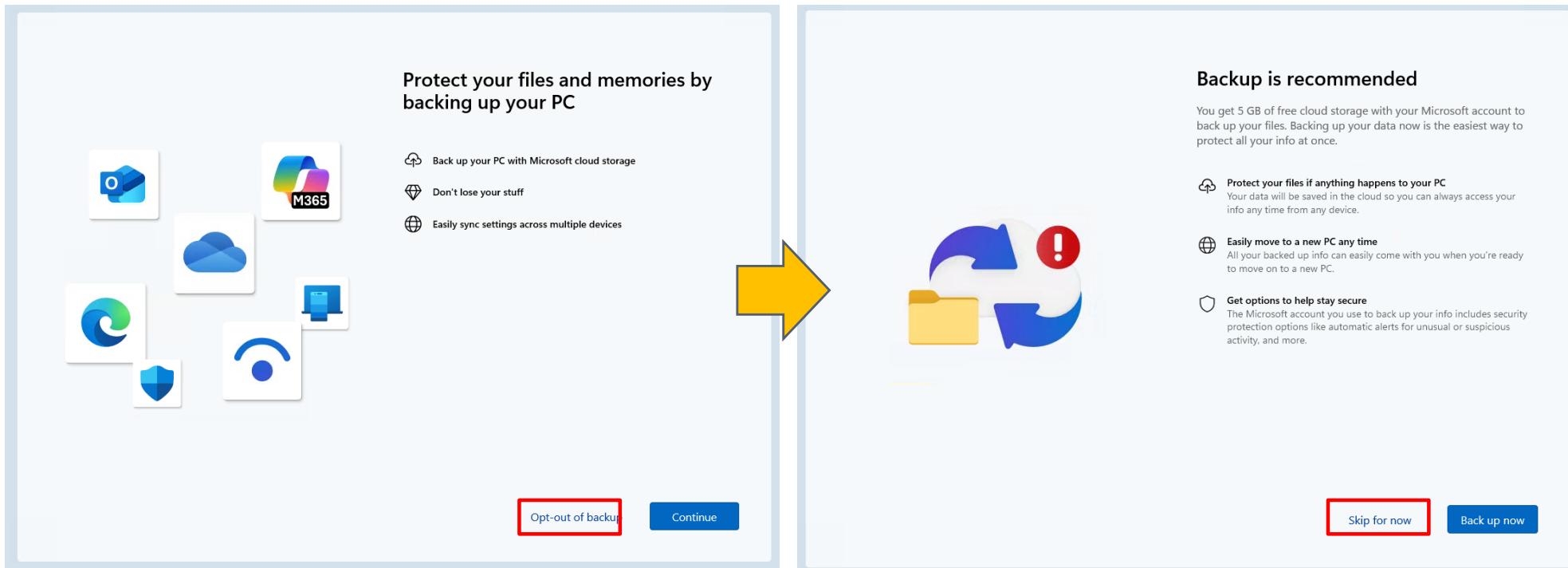
Windows マシンへのアクセス

- トポロジが表示されますので Win11 を選択します
- その後 WebRDP をクリックします



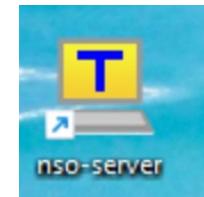
バックアップが表示された場合

- ・バックアップ画面が表示された場合は Opt-out of backup -> Skip for now を選択します



TeraTerm の起動

- デスクトップにある TeraTerm アイコンをクリックして起動します



- "docker ps" コマンドで docker が起動していることを確認してください

```
198.18.133.101 - nso@localhost: ~ VT
File Edit Setup Control Window KanjiCode Help
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 6.2.0-32-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

806 updates can be applied immediately.
805 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Wed Jul  9 07:50:17 2025 from 198.18.133.100
-bash: /home/nso/NSO-6.1.10/ncsrc: No such file or directory
nso@localhost: $ docker ps
CONTAINER ID   IMAGE          COMMAND           CREATED          STATUS          PORTS          NAMES
7345cd3fb5df  alpine:3.15    "/bin/sh -c 'ip rout..."  13 months ago   Up 14 minutes   dest
dd36859405ff  ios-xr/xrd-control-plane:7.8.1  "/bin/sh -c /sbin/xr..."  13 months ago   Up 14 minutes   xr-2
8cdad67db4b2  alpine:3.15    "/bin/sh -c 'ip rout..."  13 months ago   Up 14 minutes   source
7727c983b189  ios-xr/xrd-control-plane:7.8.1  "/bin/sh -c /sbin/xr..."  13 months ago   Up 14 minutes   xr-1
nso@localhost:~$ ■
```

alpine と xrd が 2 台ずつ起動していることを確認します

今回使用するイメージについて

- 今回のハンズオンでは、無償版 NSO イメージを使用しています
- コンテナ環境をご用意いただくことでお客様自身の環境でも同じことができますのでぜひお試しください
- イメージは下記と同じものを dCloud 上にダウンロード済みです
<https://software.cisco.com/download/home/286331591/type/286283941/release/6.5>

Software Download

The screenshot shows the Cisco Software Download interface for the Crosswork Network Services Orchestrator Free Trial. On the left, there's a sidebar with a search bar, 'Expand All' and 'Collapse All' buttons, and dropdown menus for 'Latest Release' (set to 6.5) and 'All Release' (set to 6.5). The main content area displays the 'Crosswork Network Services Orchestrator Free Trial' page for Release 6.5. It includes a 'Release 6.5' section with a 'My Notifications' button, and 'Related Links and Documentation' sections for 'NSO Dev Center'. Below this, a 'File Information' table lists two files: 'Cisco NSO Cisco ASA NED' (ncs-6.5-cisco-asa-6.18.23-freetrial.signed.bin) and 'Cisco NSO Cisco IOS NED' (ncs-6.5-cisco-ios-6.109.4-freetrial.signed.bin), along with their release dates (15-May-2025 and 14-May-2025), sizes (7.75 MB and 65.07 MB), and download links.

File Information	Release Date	Size	Action
Cisco NSO Cisco ASA NED ncs-6.5-cisco-asa-6.18.23-freetrial.signed.bin Advisories	15-May-2025	7.75 MB	
Cisco NSO Cisco IOS NED ncs-6.5-cisco-ios-6.109.4-freetrial.signed.bin Advisories	14-May-2025	65.07 MB	

NSO コンテナの紹介

NSO コンテナの利点

- NSO は現在下記のインストール方法を利用可能です

インストールファイル	インストールの種類	用途
nso-6.5-freetrial.linux.x86_64.signed.bin	ローカルインストール またはシステムインストール用	<ul style="list-style-type: none">• コンテナを利用しない NSO の利用• NSO のパッケージ開発
nso-6.5-freetrial.container-image- prod .linux.x86_64.signed.bin	コンテナ商用イメージ	<ul style="list-style-type: none">• 商用 NSO 起動• パッケージの作成、make などは可能です
nso-6.5-freetrial.container-image- build .linux.x86_64.signed.bin	開発用コンテナイマーク ※ NSO は起動しません	<ul style="list-style-type: none">• NSO パッケージのビルド (C コンパイラ、Perl 等含む)

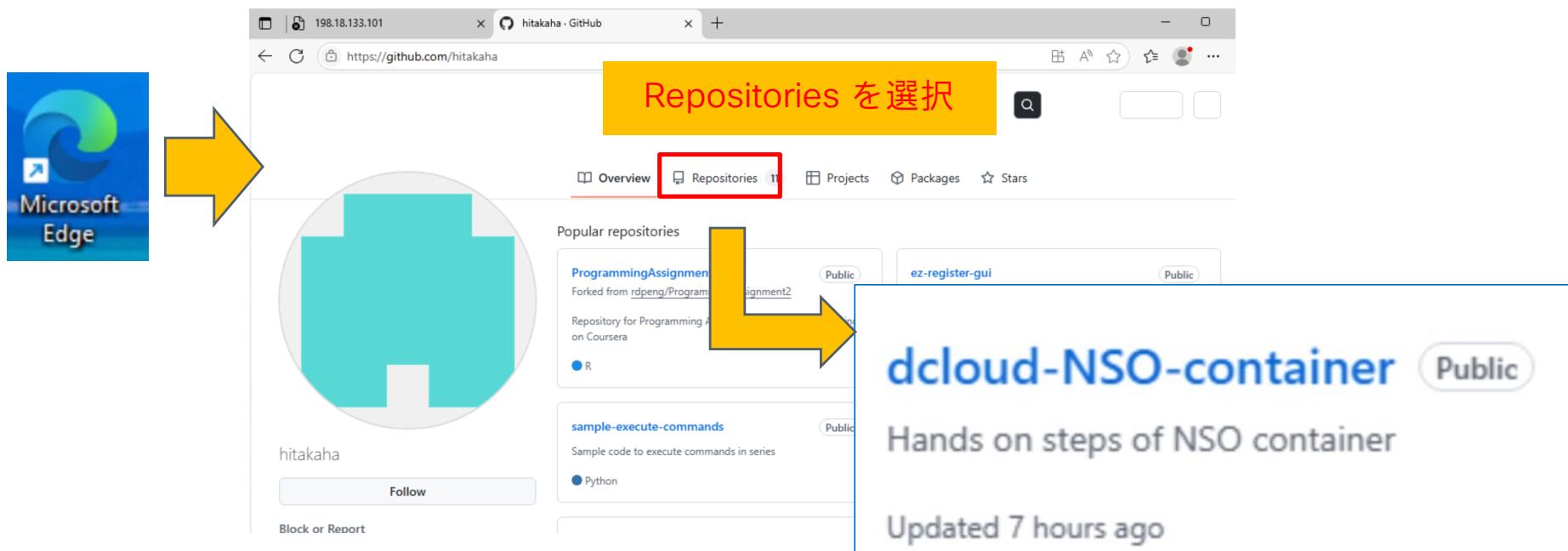
<https://nso-docs.cisco.com/guides/administration/installation-and-deployment/containerized-nso>

NSO コンテナの利点

- これまでのローカルインストール・システムインストールでも NSO をご利用いただけます
- しかし、その場合下記のような環境問題を意識する必要がありました
 - インストールするディストリビューション、カーネルのバージョン
 - SysmteV と systemd による起動問題
 - JRE/JDK, Python のバージョン依存
- コンテナ版では上記の問題を考慮する必要はありません

コピペ用 GitHub のご案内

- ・今回のシナリオをコピペできるよう、コマンド集を GitHub で公開しております
- ・dCloud 上のブラウザから <https://github.com/hitakaha> へアクセスいただき dcloud-NSO-container レポジトリをご参照ください



NSO コンテナの利用と起動

NSO コンテナ商用イメージの展開

- まず初めに NSO の prod イメージを展開します
- 展開されたファイルが分からなくならないよう、work フォルダを作りそこに展開します

```
nso@localhost:~$  
nso@localhost:~$ cd NSO-6.5-free/  
nso@localhost:~/NSO-6.5-free$ mkdir work  
nso@localhost:~/NSO-6.5-free$ cd work  
nso@localhost:~/NSO-6.5-free/work$ bash ../nso-6.5-freetrial.container-image-prod.linux.x86_64.signed.bin  
Unpacking...  
Verifying files...  
Retrieving CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...  
Successfully retrieved and verified crcam2.cer.  
Retrieving SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...  
Successfully retrieved and verified innerspace.cer.  
Successfully verified root, subca and end-entity certificate chain.  
(略)
```

NSO コンテナ商用イメージの読み込み

- 次にイメージを docker に読み込みます

```
nso@localhost:~/NSO-6.5-free/work$ docker load -i nso-6.5.container-image-prod.linux.x86_64.tar.gz
5af106f275cf: Loading layer [=====] 224.7MB/224.7MB
dcf1413174a6: Loading layer [=====] 1.407GB/1.407GB
Loaded image: cisco-nso-prod:6.5
```

```
nso@localhost:~/NSO-6.5-free/work$ docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
cisco-nso-prod      6.5      799772f04d48   8 weeks ago   1.59GB
alpine              3.15     32b91e3161c8   19 months ago  5.59MB
ios-xr/xrd-control-plane 7.8.1    7bb8619badbb   2 years ago   1.14GB
nso@localhost:~/NSO-6.5-free/work$
```

NSO コンテナ商用イメージの起動

- 最後に、admin ユーザの作成とイメージの起動を行います
- admin ユーザは ADMIN_USERNAME および ADMIN_PASSWORD を環境変数として渡すことで作成されますが、ADMIN_USERNAME=admin がデフォルトなので、そのままでも良い場合は ADMIN_PASSWORD だけ指定します

```
nso@localhost:~/NSO-6.5-free/work$ docker run -itd --name nso -e ADMIN_PASSWORD=admin cisco-ns0-prod:6.5  
b62e3845cbf16fb9020c96c57b02bc276b9d55d9642ecc6032eb3b325f312fd3  
nso@localhost:~/NSO-6.5-free/work$  
nso@localhost:~/NSO-6.5-free/work$ docker exec -it nso ncs_cli -C -u admin  
  
admin connected from 127.0.0.1 using console on b62e3845cbf1  
admin@ncs# ?  
Possible completions:  
  aaa                      AAA management  
  alarms                   Alarm management  
  autowizard               Automatically query for mandatory elements  
  call-home                Set satellite URL for Smart Licensing  
  cd                       Change working directory  
  clear                    Clear parameter  
  
(略)  
admin@ncs# exit  
nso@localhost:~/NSO-6.5-free/work$
```

? を入力しコマンドがたくさん表示されていたら成功です。
失敗すると exit と quit のみ表示されます（次項参照）

(失敗例) NSO コンテナ商用イメージの起動

- ADMIN_PASSWORD を環境変数として渡さずに起動すると admin ユーザが作成されません
- そのため ncs_cli を実行してもコマンドが一切実行されないため注意が必要です

```
nso@localhost:~/NSO-6.5-free/work$ docker run -itd --name nso cisco-nsos-prod:6.5  
b62e3845cbf16fb9020c96c57b02bc276b9d55d9642ecc6032eb3b325f312fd3  
nso@localhost:~/NSO-6.5-free/work$  
nso@localhost:~/NSO-6.5-free/work$ docker exec -it nso ncs_cli -C -u admin  
  
admin connected from 127.0.0.1 using console on 04ec5b0a1de7  
admin@ncs# ?  
Possible completions:  
  exit  Exit the management session  
  quit  Exit the management session  
admin@ncs#  
admin@ncs# exit  
nso@localhost:~/NSO-6.5-free/work$ docker stop nso  
donoso  
nso@localhost:~/NSO-6.5-free/work$ docker rm nso  
nso  
nso@localhost:~/NSO-6.5-free/work$
```

admin ユーザの作成に失敗すると exit と quit 以外実行できません
そのため前項のように忘れずに ADMIN_PASSWORD を環境変数オプション -e で指定します

NSO 6.5 での変更点

- NSO 6.4 までのコンテナイメージでは、コンテナ内は root での system-install でした
- NSO 6.5 から --run-as-user オプションを使った “nso” ユーザでの system-install に変更されています
- NSO 6.4 までのコンテナイメージをご利用の方はファイルへのアクセス権など変更になりますのでご注意ください

NSO-6.5

NSO is now installed with the '--run-as-user' option for build and production containers to run NSO from the non-root 'nso' user that belongs to the 'nso' user group. When migrating from previous versions, ensure the 'nso' user owns or has access rights to the required files and directories.

Examples include application directories, SSH host keys, SSH keys used to authenticate with devices, etc. See the deployment example variant referenced by the examples.ncs/getting-started/netsim-sshkey/README.md for an example.

The NSO container will now run a script called 'take-ownership.sh' as part of its startup, which takes ownership of all the directories that NSO needs. The script will be one of the first things to run. The script can be overridden to take ownership of even more directories, such as bind mounts or mounted volumes.

[ENG-34586/container non-backwards compatible enhancement]

NSO コンテナ商用イメージのフォルダ

- コンテナで bash を起動し、コンテナ内の主要なファイルとフォルダを理解しておきます（下記一部省略しています）

```
nso@localhost:~/NSO-6.5-free/work$ docker exec -it nso bash
[nso@c8ce3af5639a /]$ ls -l
total 92
dr-xr-xr-x  1 root root 4096 Jun 25  2024 afs
lrwxrwxrwx  1 root root    7 Jun 25  2024 bin -> usr/bin
dr-xr-xr-x  1 root root 4096 Jun 25  2024 boot
drwxr-x---  2 root nso  4096 May  6 11:06 defaults          << 初期 ncs.conf などが格納されています
lrwxrwxrwx  1 root root    7 Jun 25  2024 lib -> usr/lib
lrwxrwxrwx  1 root root    9 Jun 25  2024 lib64 -> usr/lib64
drwxr-xr-x  1 nso  root 4096 Jul  9 12:15 log             << ログ、保存する場合はここにマウントします
drwx-----  1 nso  nso  4096 Jul  9 12:16 nso            << NSO のメイン、パッケージなどはここにマウントします
dr-xr-x---  1 root root 4096 May  1 09:59 root
drwxr-xr-x  1 nso  root 4096 Jul  9 12:15 run
-r-xr-xr-x  1 root root 9227 May  6 11:06 run-nso.sh        << 初期起動スクリプト、起動時の理解に役立ちます
lrwxrwxrwx  1 root root    8 Jun 25  2024 sbin -> usr/sbin
drwxr-xr-x  1 root root 4096 Jun 25  2024 srv
-r-xr-xr-x  1 root root  968 May  6 11:06 start-crond.sh
dr-xr-xr-x  13 root root   0 Jul  9 12:15 sys
-r-xr-xr-x  1 root root  296 May  6 11:06 take-ownership.sh << マウントされたファイルのパーミッションを取得するスクリプト
drwxrwxrwt  1 root root 4096 Jul  9 12:15 tmp
drwxr-xr-x  1 root root 4096 Apr 28 15:33 usr
drwxr-xr-x  1 root root 4096 Apr 28 15:33 var
[nso@c8ce3af5639a /]$ exit
exit
nso@localhost:~/NSO-6.5-free/work$
```

NSO コンテナ商用イメージの停止と削除

- ここまで一旦確認できたら一度コンテナを停止、削除しておきます
- “docker ps -a”コマンドで正常に削除されたか確認します (-a オプションで停止中のコンテナも表示)

```
nso@localhost:~/NSO-6.5-free/work$ docker stop nso
nso
nso@localhost:~/NSO-6.5-free/work$ docker rm nso
nso
nso@localhost:~/NSO-6.5-free/work$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
7345cd3fb5df        alpine:3.15       "/bin/sh -c 'ip rout..."   13 months ago      Up About an hour   dest
dd36859405ff        ios-xr/xrd-control-plane:7.8.1    "/bin/sh -c /sbin/xr..."   13 months ago      Up About an hour   xr-2
8cdad67db4b2        alpine:3.15       "/bin/sh -c 'ip rout..."   13 months ago      Up About an hour   source
7727c983b189        ios-xr/xrd-control-plane:7.8.1    "/bin/sh -c /sbin/xr..."   13 months ago      Up About an hour   xr-1
nso@localhost:~/NSO-6.5-free/work$
```

nso がなくなっていたら成功です

ストレージのマウント

マウントの種類

- ・コンテナは停止・破棄するとデータが消えてしまいます
- ・そのため NSO では実行フォルダ /nso およびログフォルダ /log をマウントします
 - ・※最低限 /nso をマウント、 /log のマウントは任意
- ・docker でのストレージのマウントには下記の 2 種類があります
 - ・Bind mount:
 - ・ローカルのフォルダをマウントします、手軽ですがパフォーマンス面では不利です
 - ・Volume mount:
 - ・Volume を作ってマウントします、パフォーマンス面で有利ですがデータの操作が少し面倒です
- ・今回は動作理解のために、フォルダをマウントする Bind mount を使います

マウントフォルダの初期化

- ここでは下記のフォルダを作りマウントします
 - nso-root: 実行 root フォルダ → /nso にマウント
 - nso-log: ログフォルダ → /log にマウント

```
nso@localhost:~$ cd  
nso@localhost:~$ mkdir nso-root  
nso@localhost:~$ mkdir nso-log  
nso@localhost:~$ ls  
NSO-6.5-free  nso-log  nso-root  snap  
nso@localhost:~$ docker run -itd --name nso -e ADMIN_PASSWORD=admin --mount type=bind,source=/home/nso/nso-root,target=/nso --mount type=bind,source=/home/nso/nso-log,target=/log cisco-nso-prod:6.5
```

```
nso@localhost:~$ docker ps -a  
CONTAINER ID   IMAGE          COMMAND           CREATED          STATUS          PORTS     NAMES  
PORTS      NAMES  
c2c78d317c1a   cisco-nso-prod:6.5   "/run-nso.sh"  
starting)        nso  
(略)
```

次ページへ続く

NSO の再起動

- 少し時間が経過すると、起動した NSO が停止しています
- 初回起動時は必要なファイルを mount されたフォルダに生成のみ行い NSO 自体は起動しません
- そのため改めて restart することで起動が完了します

```
nso@localhost:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
c2c78d317c1a cisco-nso-prod:6.5 "/run-nso.sh" 2 minutes ago Exited (19) 2 minutes ago
nso
7345cd3fb5df alpine:3.15 "/bin/sh -c 'ip rout..." 13 months ago Up 51 minutes
dest
(略)

nso@localhost:~$
nso@localhost:~$ docker restart nso
nso@localhost:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
c2c78d317c1a cisco-nso-prod:6.5 "/run-nso.sh" 5 minutes ago Up 2 seconds (health: starting)
nso
7345cd3fb5df alpine:3.15 "/bin/sh -c 'ip rout..." 13 months ago Up 54 minutes
dest
(略)
```

マウントフォルダの確認

```
nso@localhost:~$  
nso@localhost:~$ ls nso-root/  
etc run  
nso@localhost:~$ ls nso-log/  
audit.log  ncserr.log.1    ncserr.log.siz  netconf.log          snmp.log  transerr.log  
devel.log   ncserr.log.idx  ncs.log       out-of-band-policy.log  traces   upgrade.log  
nso@localhost:~$  
nso@localhost:~$ docker exec -it nso ncs_cli -C -u admin  
admin connected from 127.0.0.1 using console on b62e3845cbf1  
admin@ncs#  
admin@ncs# ?  
Possible completions:  
  aaa           AAA management  
  alarms        Alarm management  
  autowizard    Automatically query for mandatory elements  
  call-home     Set satellite URL for Smart Licensing  
  cd            Change working directory  
admin@ncs# exit  
nso@localhost:~$
```

それぞれのフォルダにファイルが
できていたら成功です

?を入力しコマンドの候補がたくさん
出力されることを確認します

パーミッションについて

- Bind mount したフォルダの所有者等は take-ownership.sh によりコンテナ内の “nso” ユーザに置き換わります
- このユーザはホスト側のシステムには存在しないので ID で表示されています
- これらのファイルを参照することは問題ありませんが、今後書き換えなどを行う場合は必ずコンテナ経由で行います

※直接書き換えるとコンテナ側からアクセスできず NSO の動作を不安定にする恐れがあります

```
nso@localhost:~$ ls -l nso-root/
total 8
drwxr-xr-x 4 998 996 4096 Jul  9 08:31 etc
drwxr-xr-x 10 998 996 4096 Jul  9 08:31 run
nso@localhost:~$
```

ncs.confについて

- ・コンテナ版の NSO は下記の順序で ncs.conf を検出・生成します（コンテナ内 /run-nso.sh 参照）
 1. /etc/ncs/ncs.conf を参照（デフォルトでこのファイルは存在しません）
 2. 上記がなければ /nso/etc/ncs.conf を参照（/nso をマウントし外部ファイル参照にします）
 3. 上記がなければ /defaults/ncs.conf を /etc/ncs/ncs.conf にコピー
- ・すでに NSO の開発環境があり、ncs.conf がある場合には nso-root/etc にコピーします
- ・もし新規にコンテナを起動する場合、一度起動し上記 3 番でコンテナ内の /etc/ncs に生成された ncs.conf を /nso/etc にコピーすることで永続化します（今回はこちらを実施します）

ncs.conf のコピー

- この段階ではまだ ncs.conf が保存されないのでコンテナ内 /etc/ncs に生成された ncs.conf を コンテナから見た /nso/etc (=nso-root/etc) にコピーします

```
nso@localhost:~$ ls nso-root/etc  
ncs.crypto_keys  ssh  ssl  
nso@localhost:~$ docker exec -it nso bash  
bash-5.1$ cp /etc/ncs/ncs.conf /nso/etc  
bash-5.1$ exit  
nso@localhost:~$ ls nso-root/etc  
ncs.conf  ncs.crypto_keys  ssh  ssl
```

nso-root/etc に ncs.conf がないことを確認

コンテナ内 /etc/ncs から /nso/etc にコピー

nso-root/etc にコピーされ永続化されました

NED のインストール

NED の展開

- コンテナイメージ同様に work フォルダに NED のファイルを展開します

```
nso@localhost:~$ cd  
nso@localhost:~$ cd NSO-6.5-free/work/  
nso@localhost:~/NSO-6.5-free/work$ bash ../ncs-6.5-cisco-iosxr-7.69-freetrial.signed.bin  
Unpacking...  
nso@localhost:~/NSO-6.5-free/work$ ls  
cisco_x509_verify_release.py3  
4.2.11.tar.gz.signature  
ncs-6.5-cisco-iosxr-7.69.tar.gz  
ncs-6.5-cisco-iosxr-7.69.tar.gz.signature  
6.0.6.tar.gz.signature  
ncs-6.5-observability-exporter-1.5.0.tar.gz  
prod.linux.x86_64.tar.gz  
ncs-6.5-observability-exporter-1.5.0.tar.gz.signature  
prod.linux.x86_64.tar.gz.signature  
ncs-6.5-phased-provisioning-1.2.0.tar.gz  
ncs-6.5-phased-provisioning-1.2.0.tar.gz.signature  
ncs-6.5-resource-manager-project-4.2.11.tar.gz  
nso@localhost:~/NSO-6.5-free/work$  
  
ncs-6.5-resource-manager-project-  
ncs-6.5-tailf-hcc-project-6.0.6.tar.gz  
ncs-6.5-tailf-hcc-project-  
nso-6.5.container-image-  
nso-6.5.container-image-  
README.signature  
tailf.cer
```

NED のインストール

- 展開した NED ファイルを docker cp コマンドでコピーします

```
nso@localhost:~/NSO-6.5-free/work$ docker cp ncs-6.5-cisco-iosxr-7.69.tar.gz nso:/nso/run/packages
Successfully copied 51.5MB to nso:/nso/run/packages
nso@localhost:~/NSO-6.5-free/work$ cd
nso@localhost:~$ ls nso-root/run/packages/
ncs-6.5-cisco-iosxr-7.69.tar.gz
nso@localhost:~$
```

パッケージのリロード

- ・パッケージのリロードを行い、result が true になったら成功です

```
nso@localhost:~$ docker exec -it nso ncs_cli -C -u admin

User admin last logged in 2025-07-07T13:40:56.03706+00:00, to c9e0a4c4b8a3, from 127.0.0.1 using cli-
console
admin connected from 127.0.0.1 using console on c9e0a4c4b8a3
admin@ncs# packages reload

>>> System upgrade is starting.
>>> Sessions in configure mode must exit to operational mode.
>>> No configuration changes can be performed until upgrade has completed.
>>> System upgrade has completed successfully.
reload-result {
    package cisco-iosxr-cli-7.69
    result true
}
admin@ncs#
```

デバイスの登録

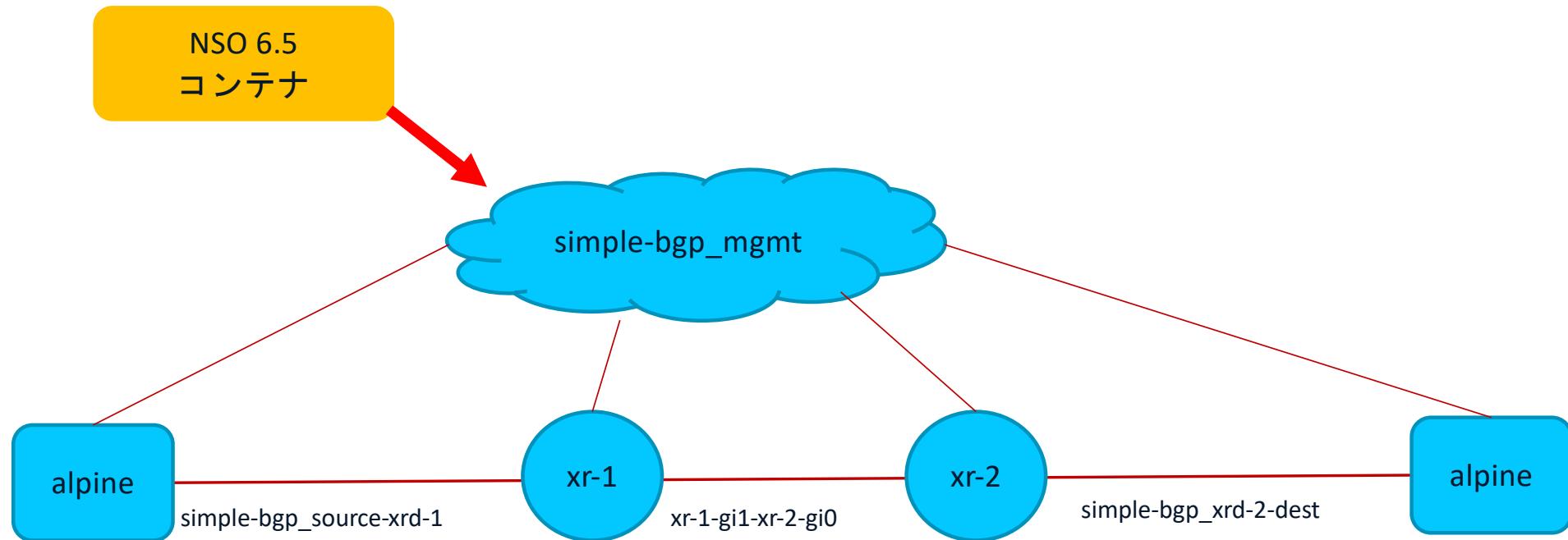
- XRd デバイスを登録します

```
admin@ncs# config t
Entering configuration mode terminal
admin@ncs(config)# devices authgroups group XR umap admin remote-name admin remote-password cisco123
remote-secondary-password cisco123
admin@ncs(config-umap-admin)# top
admin@ncs(config)# devices device xr-1 address 172.30.0.2 device-type cli protocol ssh ned-id cisco-
iosxr-cli-7.69
admin@ncs(config-device-xr-1)# authgroup XR
admin@ncs(config-device-xr-1)# state admin-state unlocked
admin@ncs(config-device-xr-1)# top
admin@ncs(config)# devices device xr-2 address 172.30.0.3 device-type cli protocol ssh ned-id cisco-
iosxr-cli-7.69
admin@ncs(config-device-xr-2)# authgroup XR
admin@ncs(config-device-xr-2)# state admin-state unlocked
admin@ncs(config-device-xr-2)# commit
Commit complete.
admin@ncs(config-device-xr-2)# end
admin@ncs#
admin@ncs# exit
exit
nso@localhost:~$
```

ネットワークへの接続

- XRd の docker network は下記のようになっていますのでそこに繋ぎます

```
nso@localhost:~$ docker network connect simple-bgp_mgmt nso
```



SSH ホストキーの取得

- 次に SSH ホストキーを取得します

```
nso@localhost:~$ docker exec -it nso ncs_cli -C -u admin

User admin last logged in 2025-07-08T06:41:45.57598+00:00, to 48ad9e916ce1, from 127.0.0.1 using cli-console
admin connected from 127.0.0.1 using console on 48ad9e916ce1
admin@ncs# devices fetch-ssh-host-keys
fetch-result {
    device xr-1
    result updated
    fingerprint {
        algorithm ssh-ed25519
        value a3:f4:b7:f4:69:2d:36:65:d0:ad:e8:e6:c3:87:ee:09
    }
}
fetch-result {
    device xr-2
    result updated
    fingerprint {
        algorithm ssh-ed25519
        value d7:13:ce:5e:9e:7f:45:62:37:0d:fd:b2:18:8d:9e:cd
    }
}
admin@ncs#
```

デバイスの sync-from

- sync-from ができたなら成功です

```
admin@ncs# devices sync-from
sync-result {
    device xr-1
    result true
}
sync-result {
    device xr-2
    result true
}
admin@ncs#
```

NSO WebUI、RESTCONF の 有効化

ncs.conf を書き換え

- WebUI および RESTCONF を使うために下記の 3 つを ncs.conf で変更します
 - Local Authentication の有効化: これで admin/admin でログインできます
 - server-alias の追加 デフォルトでは localhost のみのためアドレスを追加します
 - WebUI の有効化

```
nso@localhost:~$ docker exec -it nso bash  
bash-5.1$ chmod 644 /nso/etc/ncs.conf  
bash-5.1$ vi /nso/etc/ncs.conf
```

```
281 <local-authentication>  
282   <enabled>false</enabled>  
283 </local-authentication>  
284  
285
```

```
328 <webui>  
329   <enabled>true</enabled>  
330   <transport>  
331     <tcp>  
332       <enabled>false</enabled>  
333       <ip>0.0.0.0</ip>  
334       <port>8080</port>  
335     </tcp>
```

Vi Tips:

- :set number で行番号表示
- i で挿入モード
- ESC で通常モード
- :wq で書き込み&終了

```
281 <local-authentication>  
282   <enabled>true</enabled>  
283 </local-authentication>  
284  
285
```

```
328 <webui>  
329   <enabled>true</enabled>  
330   <server-alias>198.18.133.101</server-alias>  
331   <transport>  
332     <tcp>  
333       <enabled>true</enabled>  
334       <ip>0.0.0.0</ip>  
335       <port>8080</port>
```

追加

```
bash-5.1$ exit  
nso@localhost:~$
```

NSO 6.5 からの変更点

- NSO 6.5 からホスト名以外への WebUI, RESTCONF のアクセスできなくなりました
- <https://nso-docs.cisco.com/guides/whats-new#web-server-hostname-matching>
- デフォルトのホスト名 localhost 以外でのアクセスを許可するには server-alias タグで追記をします

▼ Web Server Hostname Matching

NSO supports serving web traffic from multiple domains and IP addresses. This functionality is configured by `server-name` and `server-alias` settings in the `ncs.conf` file. In addition, the web server refuses to serve requests to other domain names and addresses by default, in order to not expose the system to redirect-related attacks. This functionality can be disabled, but that is strongly discouraged.

<https://nso-docs.cisco.com/guides/whats-new#web-server-hostname-matching>

NSO コンテナ商用イメージの再起動

- 一度コンテナ NSO を停止
- RESTCONF/WebUI 用ポートを追加して起動します

```
nso@localhost:~$ docker stop nso
nso
nso@localhost:~$ docker rm nso
nso
nso@localhost:~$
nso@localhost:~$ docker run -itd --name nso -e ADMIN_PASSWORD=admin --mount
type=bind,source=/home/nso/nso-root,target=/nso --mount type=bind,source=/home/nso/nso-log,target=/log
-p 8080:8080 cisco-nso-prod:6.5
b62e3845cbf16fb9020c96c57b02bc276b9d55d9642ecc6032eb3b325f312fd3
nso@localhost:~$ docker network connect simple-bgp_mgmt nso
nso@localhost:~$ docker exec -it nso ncs_cli -C -u admin
admin connected from 127.0.0.1 using console on b62e3845cbf1
admin@ncs# ?
(略)                                     NSO CLIへのアクセスとコマンド候補が出力されることを確認
admin@ncs# exit
nso@localhost:~$
```

動作確認

- RESTCONF および Web ブラウザからのアクセス確認

```
nso@localhost:~$ curl -u admin:admin http://198.18.133.101:8080/restconf
<restconf xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
  <data/>
  <operations/>
  <yang-library-version>2019-01-04</yang-library-version>
</restconf>
nso@localhost:~$
```

The screenshot shows the NSO web interface. The top navigation bar includes a Cisco logo, the title "Crosswork Network Services Orchestrator", and a user dropdown set to "admin". A sidebar on the left has links for "Home", "Devices", "Services", "Config editor", and "Tools". The main content area is titled "Home" and displays a warning message: "Your NSO license has expired or is not sufficient for the current setup. For more information about your company's licensing account, go [here](#)". Below the message are two cards: "Devices" and "Services". A yellow callout box on the right side contains the URL and credentials used for the RESTCONF request: "http://198.18.133.101:8080/", "username=admin", and "password=admin".

WebUI での sync-from

- Devices から全デバイスを選び、Sync-from ができたたら完成です！

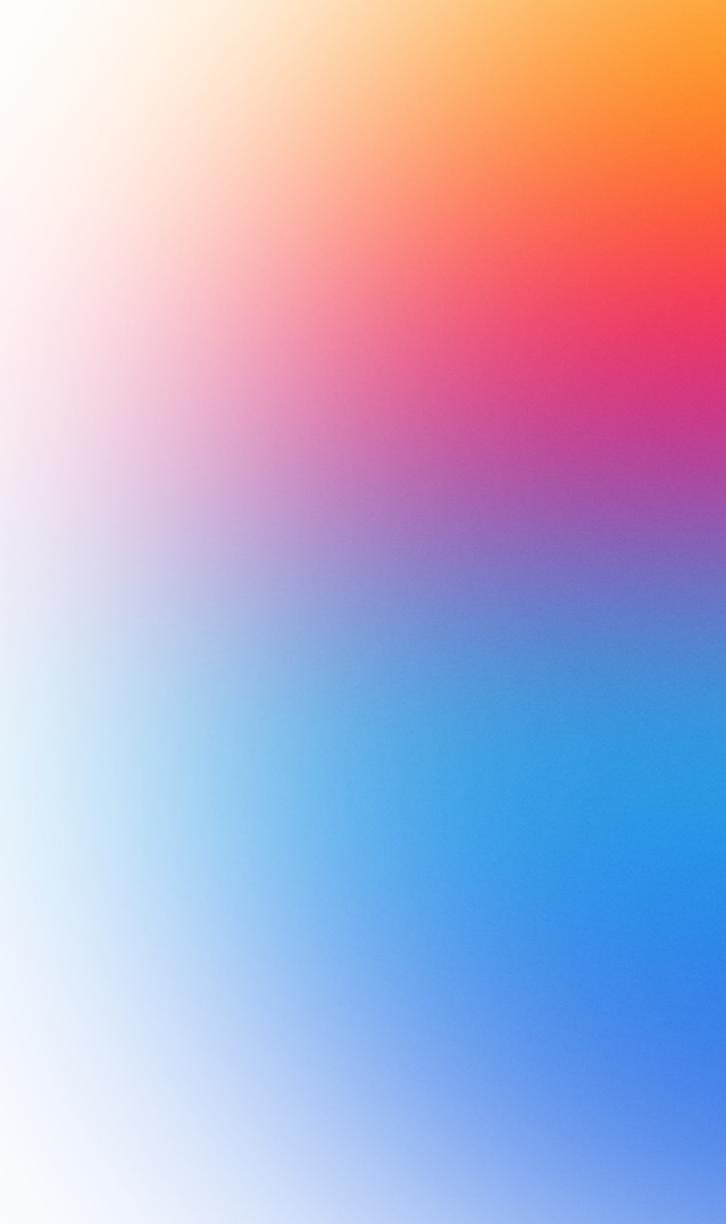
The screenshot shows the Crosswork Network Services Orchestrator WebUI. The left sidebar has 'Devices' selected. A modal dialog titled 'Action: Sync from' is open, showing the following content:

Number of device(s):
2

Result

```
sync-result {  
    device xr-1  
    result true  
}  
sync-result {  
    device xr-2  
    result true  
}
```

At the bottom of the dialog are 'Done' and 'Cancel' buttons. In the background, the main interface shows a list of devices with 'xr-1' and 'xr-2' listed, and a 'Device group' section.



余裕のある方向けシナリオ

ゴール

- 下記 github に公開されている bgpmgr をダウンロードして動かすことを目標にします

The screenshot shows the GitHub repository page for `dcloud-bgpmgr`. The repository is public and was created by `hitakaha`. It contains one branch and no tags. The main commit is from `hitakaha` and is labeled "first commit". The commit message is "Configure iBGP/eBGP between two IOS-XR routers". The repository includes several files and folders: `load-dir`, `python/bgpmgr`, `src`, `templates`, `test`, `README`, `README.md`, and `package-meta-data.xml`. The `README` file is currently selected. The repository has 0 stars, 0 forks, and 0 releases published.

git clone とコピー

```
nso@localhost:~$ git clone https://github.com/hitakaha/dcloud-bgpmgr.git
Cloning into 'dcloud-bgpmgr'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 31 (delta 3), reused 31 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (31/31), 20.47 KiB | 403.00 KiB/s, done.
Resolving deltas: 100% (3/3), done.
nso@localhost:~$
nso@localhost:~$ docker cp dcloud-bgpmgr/ nso:/nso/run/packages
Successfully copied 132kB to nso:/nso/run/packages
nso@localhost:~$
```

デバイスからの BGP 削除

```
nso@localhost:~$ docker exec -it nso ncs_cli -C -u admin

User admin last logged in 2025-07-08T13:02:04.604421+00:00, to 98de9250eb32, from 127.0.0.1 using cli-
console
admin connected from 127.0.0.1 using console on 98de9250eb32
admin@ncs# config t
Entering configuration mode terminal
Current configuration users:
admin console (cli from 127.0.0.1) on since 2025-07-08 13:01:01 terminal mode
admin@ncs(config)# devices device xr-1 config
admin@ncs(config-config)# no router bgp
admin@ncs(config-config)# top
admin@ncs(config)# devices device xr-2 config
admin@ncs(config-config)# no router bgp
admin@ncs(config-config)# commit
Commit complete.
admin@ncs(config-config)# end
admin@ncs#
admin@ncs# show running-config devices device * config router bgp
% No entries found.
admin@ncs#
```

パッケージリロード

```
admin@ncs# packages reload
>>> System upgrade is starting.
>>> Sessions in configure mode must exit to operational mode.
>>> No configuration changes can be performed until upgrade has completed.
>>> System upgrade has completed successfully.
reload-result {
    package bgpmgr
    result true
}
reload-result {
    package cisco-iosxr-cli-7.69
    result true
}
admin@ncs#
```

動作確認

```
admin@ncs# config t
Entering configuration mode terminal
admin@ncs(config)# bgpmgr test
admin@ncs(config-bgpmgr-test)# dev1 xr-1
admin@ncs(config-bgpmgr-test)# dev1-as 100
admin@ncs(config-bgpmgr-test)# dev2 xr-2
admin@ncs(config-bgpmgr-test)# dev2-as 200
admin@ncs(config-bgpmgr-test)# commit
Commit complete.
admin@ncs(config-bgpmgr-test)# end
admin@ncs# devices device * live-status exec any show bgp summary
devices device xr-1 live-status exec any
    result
(略)
Neighbor      Spk      AS MsgRcvd MsgSent      TblVer  InQ OutQ Up/Down St/PfxRcd
10.2.1.3        0     200        4      4          5      0    0 00:00:10           3
RP/0/RP0/CPU0:xr-1#
devices device xr-2 live-status exec any
(略)
Neighbor      Spk      AS MsgRcvd MsgSent      TblVer  InQ OutQ Up/Down St/PfxRcd
10.2.1.2        0     100        4      4          5      0    0 00:00:10           3
RP/0/RP0/CPU0:xr-2#
admin@ncs#
```

動作確認 (WebUI)

The screenshot shows the Crosswork Network Services Orchestrator web interface. The URL in the browser is <https://198.18.133.101:8080/webui-one/Services/Details/%2Fbgpmgr%3Abgpmgr%7Btest%7D/Details>. The page title is "Crosswork Network Services Orchestrator". The main content area displays a service named "test" with a service type of "/bgpmgr:bgpmgr". There are two tabs: "Details" (selected) and "Log". The "Details" tab shows the following data:

Modified devices	2 device(s)	Directly-modified devices	2 device(s)
Modified services	0 service(s)	Directly-modified services	0 service(s)

The left sidebar includes navigation links for Home, Devices, Services (which is selected), Config editor, and Tools.

アンケートのお願い

今後のハンズオンで取り上げて欲しいテーマに投票をお願いします（複数可）

- YANG フル活用、 union, enumerate, deref() の操作理解や CDB を拡張してみよう！
- XML テンプレート言語を使ってみよう！
- 自分だけのコマンドを作れる、“Action” を作ってみよう！
- 階層型サービスモデル Stacked Service を作ってみよう！
- 変更を監視する “kicker” や進捗を管理する “Plan” でワークフローを動かそう！
- NanoService を使ってみよう！
- NSO 関連便利ツール体験 (Resource Manager, Observability exporter etc.)
- NSO HA を触ってみよう！
- その他



<https://forms.office.com/r/GGv8qVZt98>

上記にアクセスいただき “Start Now”
もしくは「今すぐ開始」をクリックして
ご回答をお願いいたします

