
Add-on アプリケーション開発 - 環境構築マニュアル - リリース 1.1

サン電子株式会社 **M2M** 事業部 ソフトウェア開発部

2019年09月17日

目次

第 1 章	はじめに	1
1.1	取り扱う話題	1
1.2	対象読者	1
1.3	必要なもの	2
第 2 章	開発者モード	3
2.1	開発者モード OFF 時の制限	3
2.2	開発者モード ON 時の制限の除去と緩和	3
第 3 章	Linux ディストリビューションに SDK をインストールする	5
第 4 章	Linux ディストリビューションにコマンドをインストールする	7
4.1	Debian GNU/Linux 8 の場合, Debian GNU/Linux 9, Ubuntu 16.04 LTS の場合	7
第 5 章	Linux ディストリビューションに Google Chrome ブラウザをインストールする	9
5.1	Windows 10 Pro の WSL 機能を使って Ubuntu を利用している場合	9
5.2	Debian GNU/Linux 8, Debian GNU/Linux 9, Ubuntu 16.04 LTS の場合	14
第 6 章	NSX7000 に SSH ログインの設定をする	17
6.1	SSH ログイン用の鍵対を作成する	18
6.2	NSX7000 の Web UI を使って SSH ログイン設定を行なう	19
6.3	SSH ログインできることを確認する	29
第 7 章	NSX7000 の開発者モードを ON にする	31
7.1	開発者モードの ON/OFF を表示する	31
7.2	次回起動時に開発者モードを ON になるように設定する	32
7.3	次回起動時に開発者モードを OFF になるように設定する	32
7.4	次回起動時の開発者モードを ON になるように設定して再起動する	32
第 8 章	おわりに	33
付録 A	Ubuntu を Windows 10 Pro にインストールする	35
付録 B	開発者モード OFF 時の NSX7000 での制限	41
B.1	開発者モード OFF 時に root 権限を持ったアプリケーションでできないことの例	42

第 1 章

はじめに

この文書では NSX7000 用 Add-on アプリケーション開発環境を構築するための手順について説明します。
ファームウェア・バージョン 1.3.1 以降がインストールされた NSX7000 を対象とします。

1.1 取り扱う話題

この文書では下記的话题を取り扱います。

- NSX7000 の開発者モード
- Linux ディストリビューションに SDK をインストールする手順
- Linux ディストリビューションにコマンドをインストールする手順
- Linux ディストリビューションに Google Chrome ブラウザをインストールする手順
- NSX7000 に SSH ログインの設定をする手順
- NSX7000 の開発者モードを ON にする手順

1.2 対象読者

この文書は下記の知識・経験を有した読者を対象とします。

- 一般的な Linux ディストリビューション^{*1}での基本的なコマンドライン操作に習熟している

^{*1} 例. Debian GNU/Linux, Ubuntu, Red Hat Linux など

1.3 必要なもの

NSX7000 用 Add-on アプリケーションを開発するには一般的な Linux ディストリビューションがインストールされたホスト PC, もしくは仮想マシンが必要です.

本文書の手順は下記の Linux ディストリビューションで動作検証しています.

- Debian GNU/Linux 8(jessie)
- Debian GNU/Linux 9(stretch)
- Ubuntu 16.04 LTS(Xenial Xerus)

SDK は 64 ビット版の Linux ディストリビューションのみサポートします.

Windows 10 Pro(64 bit) を使って評価する場合, WSL(Windows Service for Linux) 機能を有効にし, Windowsストアから Ubuntu を入手してください(「*Ubuntu* を *Windows 10 Pro* にインストールする」).

第 2 章

開発者モード

NSX7000 はファームウェア・バージョン 1.3.1 から 開発者モード をサポートします。

本章ではこの開発者モードについて説明します。

2.1 開発者モード OFF 時の制限

工場出荷時, NSX7000 は 開発者モード OFF です. このとき, NSX7000 は下記の制限を課します.

- NSX7000 は署名ファイルなしの Add-on パッケージのインストールを拒否する
- NSX7000 は任意のケーパビリティ^{*1*2}を必要とする Add-on アプリケーションからのシステムコールを拒否する
- NSX7000 は SSH での root アカウントでのログインを拒否する

2.2 開発者モード ON 時の制限の除去と緩和

開発者モード ON にすることでシステムの制限を除去, または, 緩和できます.

- NSX7000 は署名ファイルなしの Add-on パッケージのインストールを許可する
- NSX7000 は `rooster-os-cap-add-on` コマンドを使うことで, ケーパビリティを必要とする Add-on アプリケーションのシステムコールを許可する (一部^{*3}を除く)
- NSX7000 は SSH での root アカウントでのログインを許可する

*1 ケーパビリティについては <http://man7.org/linux/man-pages/man7/capabilities.7.html> を参照してください.

*2 開発者モード OFF 時に制限されるケーパビリティについては (「開発者モード OFF 時の NSX7000 での制限」) を参照してください.

*3 開発者モード ON 時に制限されるケーパビリティについては (「開発者モード ON 時の NSX7000 での制限」) を参照してください.

第 3 章

Linux ディストリビューションに SDK をインストールする

Linux ディストリビューションに Add-on アプリケーション開発用 SDK をインストールしてください。

SDK は RoosterOS-SDK-x.y.z.tar.xz です。

注釈: RoosterOS-SDK- x.y.z .tar.xz の x, y, z は数字です. (例) RoosterOS-SDK-1.3.1.tar.xz

ホーム・ディレクトリに RoosterOS-SDK-x.y.z.tar.xz を展開します。

作成された RoosterOS-SDK-x.y.z ディレクトリを指すシンボリックリンク RoosterOS-SDK を作成します。

```
$ cd $HOME
$ tar xvf RoosterOS-SDK-x.y.z.tar.xz
$ ln -sf RoosterOS-SDK-x.y.z RoosterOS-SDK
```


第 4 章

Linux ディストリビューションにコマンドをインストールする

Add-on アプリケーション開発を行なうにあたり, Linux ディストリビューションに下記のコマンドが必要です. 各ディストリビューション固有の方法でこれらのコマンドが使用できるようにしてください.

- ssh
- ssh-keygen
- ssh-agent
- ssh-add
- scp
- mksquashfs
- fakeroot
- make

4.1 Debian GNU/Linux 8 の場合, Debian GNU/Linux 9, Ubuntu 16.04 LTS の場合

Debian GNU/Linux 8, Debian GNU/Linux 9, および, Ubuntu 16.04 LTS では下記のパッケージをインストールすることで前述のコマンドが使用できるようになります.

- openssh-client パッケージ
- squashfs-tools パッケージ
- fakeroot パッケージ

- make パッケージ

これらのパッケージをインストールするにはつぎのコマンドを実行します。

```
$ sudo apt install openssh-client squashfs-tools fakeroot make
```

Windows 10 Pro の WSL 機能を使って Ubuntu を利用している場合のみ, 上記のパッケージのインストール後に下記のコマンドも実行します。

```
$ sudo update-alternatives --set fakeroot /usr/bin/fakeroot-tcp
```

第 5 章

Linux ディストリビューションに Google Chrome ブラウザをインストールする

NSX7000 は Google Chrome ブラウザをサポートします。

本文書での手順の実行にあたり、Google Chrome ブラウザが必要です。各ディストリビューション固有の方法で Google Chrome ブラウザを使用できるようにしてください。

なお、Windows 10 Pro の WSL 機能を使って Ubuntu を利用している場合は、Ubuntu ではなく Windows 10 Pro に Google Chrome ブラウザをインストールします。

公式情報「Google Chrome をダウンロードしてインストールする」^{*1} も参照してください。

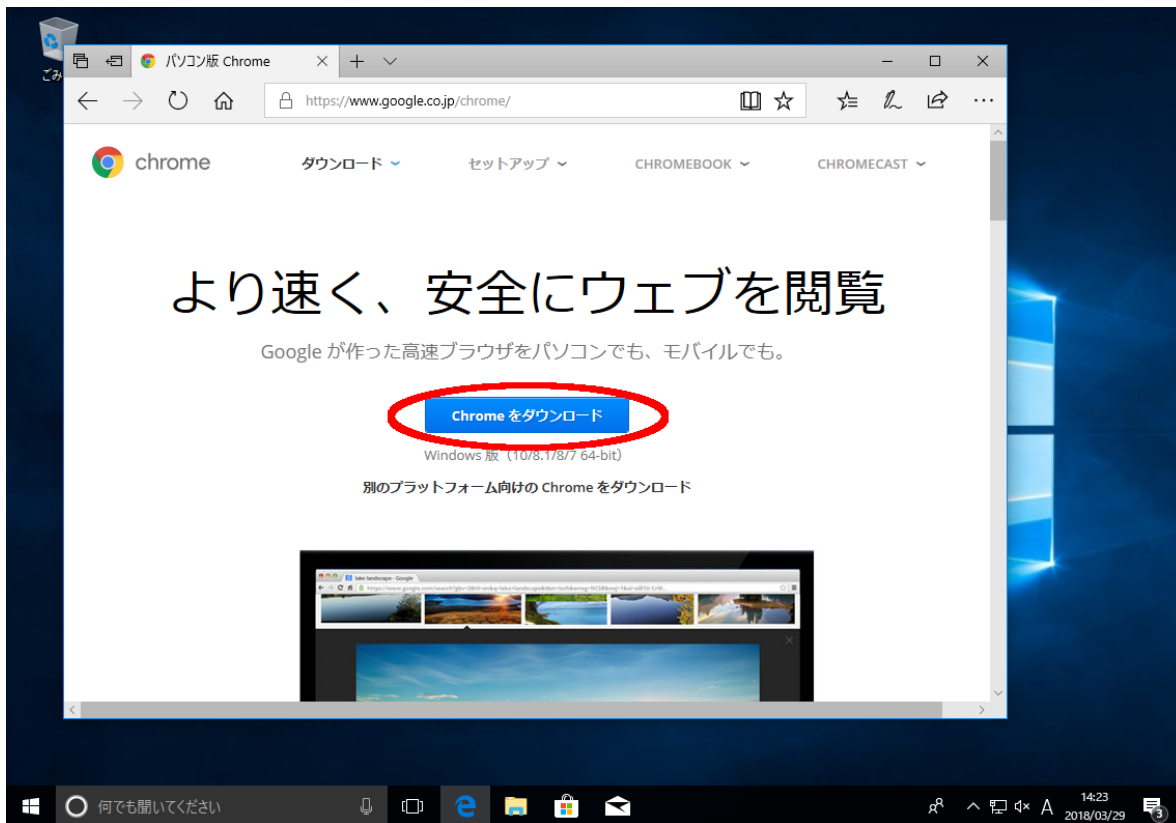
5.1 Windows 10 Pro の WSL 機能を使って Ubuntu を利用している場合

Windows 10 Pro の WSL 機能を使って Ubuntu を利用している場合、Google Chrome ブラウザのオフィシャル・サイトから Google Chrome ブラウザをダウンロードし、Windows 10 Pro に直接インストールします。

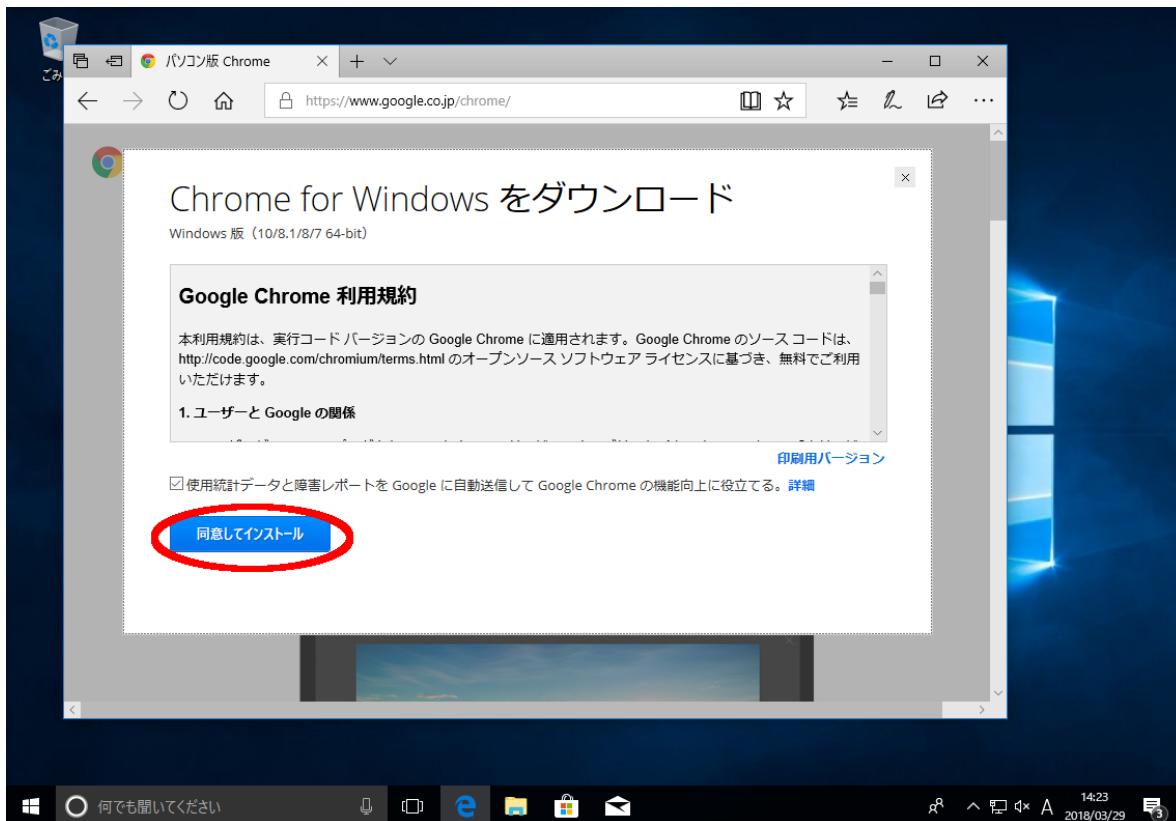
1. Microsoft Edge ブラウザを起動し、オフィシャル・サイト <https://www.google.co.jp/chrome/> にアクセスする

^{*1} https://support.google.com/chrome/answer/95346?hl=ja&ref_topic=7439538

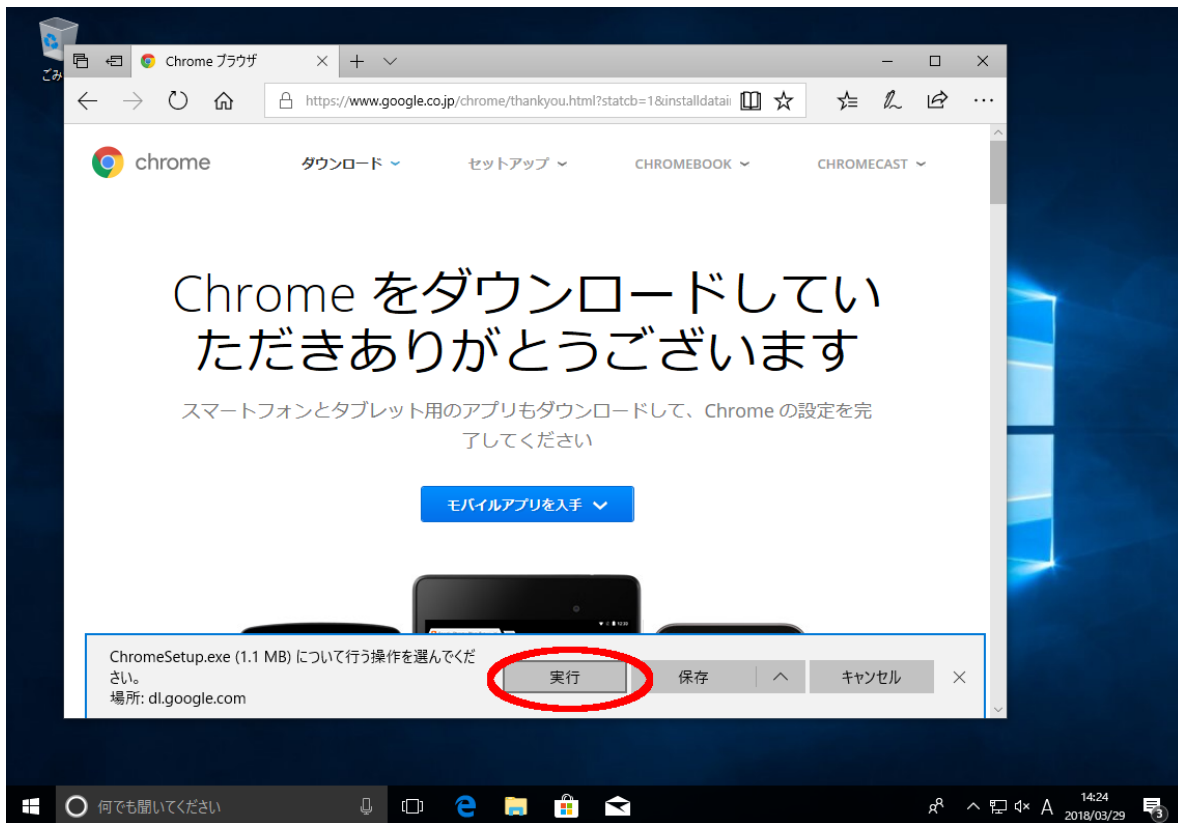
2. 「Chrome をダウンロード」ボタンをクリックする



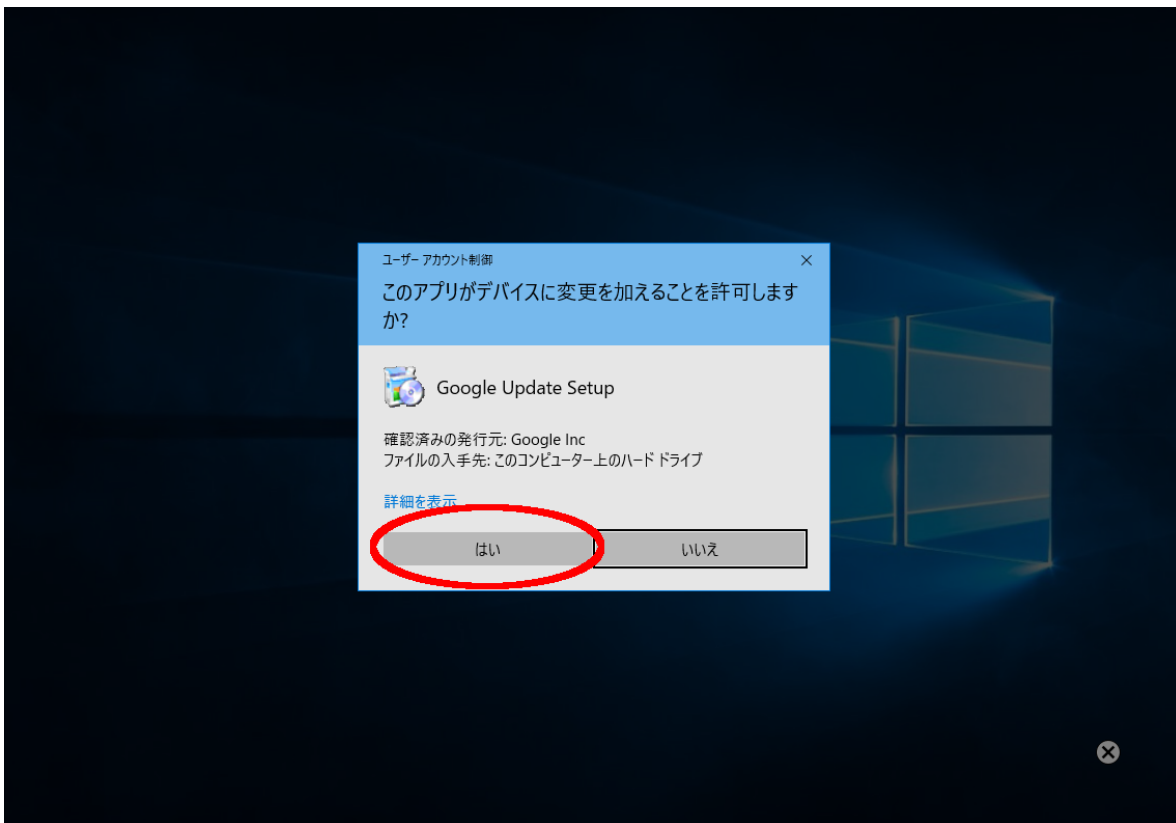
3. 「同意してインストール」ボタンをクリックする



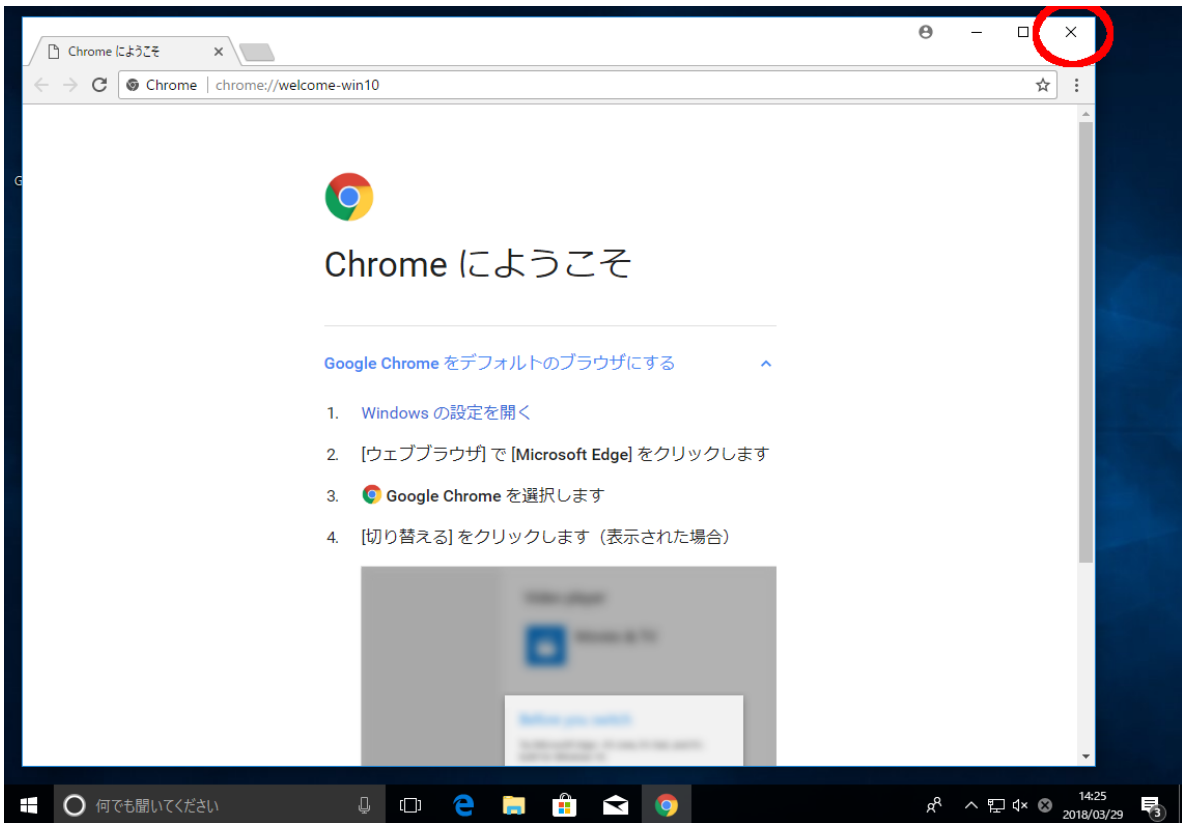
4. 「実行」ボタンをクリックする



5. 「はい」 ボタンをクリックし, Google Update Setup がデバイスに変更を加えることを許可する



6. インストールが終了し, Google Chrome が自動的に起動するのを待つ
7. インストール終了後に Google Chrome が自動起動したら, ウィンドウ右上の「X」をクリックして終了する



5.2 Debian GNU/Linux 8, Debian GNU/Linux 9, Ubuntu 16.04 LTS の場合

Debian GNU/Linux 8, Debian GNU/Linux 9, Ubuntu 16.04 LTS の場合, 下記のコマンドの実行で Google Chrome ブラウザをインストールできます.

1. Google のパッケージ署名を検証するための公開鍵をインストールする

```
$ pub_key_url=https://dl-ssl.google.com/linux/linux_signing_key.pub
$ wget -q -O - $pub_key_url | sudo apt-key add -
```

2. 設定ファイル/etc/apt/sources.list.d/my-google.list を追加し, Google のリポジトリ情報をシステムに追加する

```
$ sudo su
# google_repo=http://dl.google.com/linux/chrome/deb/
# source_list=/etc/apt/sources.list.d/my-google.list
# echo "deb [arch=amd64] $google_repo stable main" >> $source_list
# exit
```

```
$ sudo apt update
```

3. Google Chrome をインストールする

```
$ sudo apt install google-chrome-stable
```

4. 作成した設定ファイル/etc/apt/sources.list.d/my-google.list を削除する

```
$ source_list=/etc/apt/sources.list.d/my-google.list  
$ sudo rm $source_list
```


第 6 章

NSX7000 に SSH ログインの設定をする

NSX7000 は公開鍵認証による suncorp アカウントでの SSH ログインをサポートします。

NSX7000 の Web UI を使って下記の設定を行うと, suncorp アカウントで NSX7000 に SSH ログインできます。

- SSH サーバ機能を有効にする
- suncorp アカウントの公開鍵を設定する

なお, 工場出荷時の状態では suncorp アカウント以外のアカウント (root アカウントを含む) ではログインできません。開発者モードを ON にする^{*1} と root アカウントでログインできるようになります。

^{*1} 開発者モードを ON にする方法については「[NSX7000 の開発者モードを ON にする](#)」を参照してください。

6.1 SSH ログイン用の鍵対を作成する

ssh-keygen コマンドを使って SSH ログイン用の鍵対を作成します。以下の ssh-keygen コマンドの実行で \$HOME/.ssh ディレクトリ以下に下記のファイルが作成されます。

- id_rsa
- id_rsa.pub

id_rsa が秘密鍵を含むファイル, id_rsa.pub が公開鍵を含むファイルです。

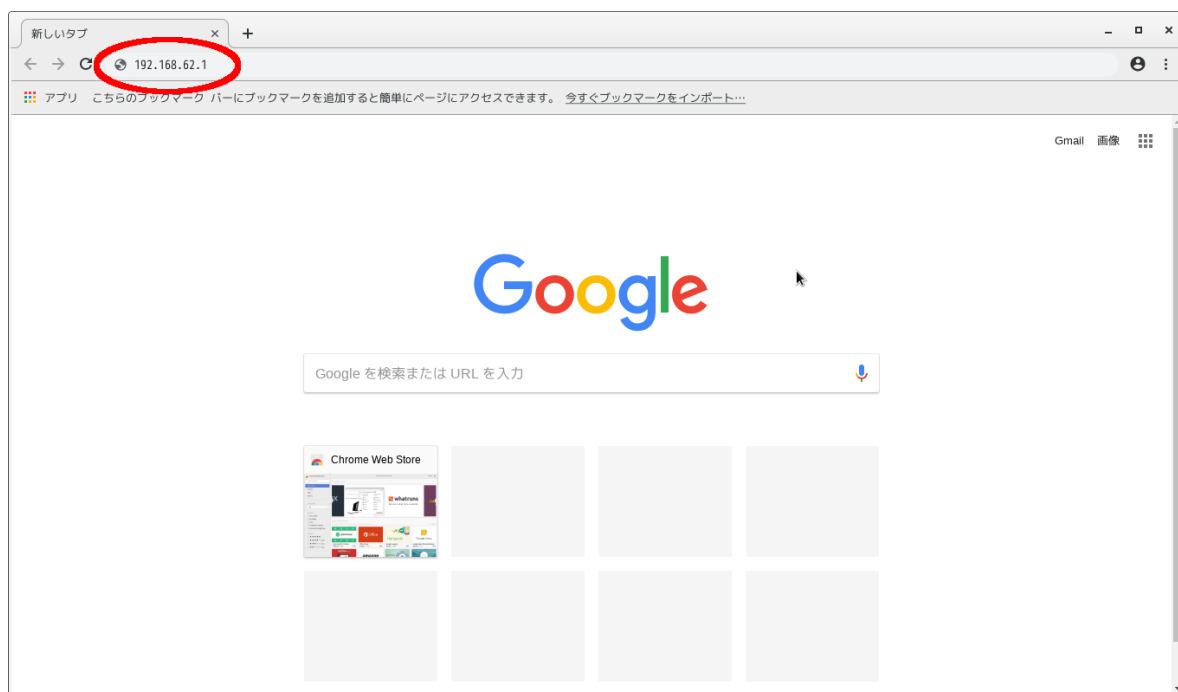
```
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/m2m/.ssh/id_rsa): <Enter を入力する>
Created directory '/home/m2m/.ssh'.
Enter passphrase (empty for no passphrase): <パスフレーズを入力する>
Enter same passphrase again: <再度パスフレーズを入力する>
Your identification has been saved in /home/m2m/.ssh/id_rsa.
Your public key has been saved in /home/m2m/.ssh/id_rsa.pub.
The key fingerprint is:
17:ab:31:6e:9a:fd:aa:1b:cf:f9:97:c7:c0:ed:fa:37 m2m@debian
The key's randomart image is:
+---[RSA 4096]-----+
|
|
|      .      |
|     o      |
|    S o . .  |
|   . = o .   |
|  . +   =    |
|   O . o +E. |
|  =o*+o..+...|
+-----+

```

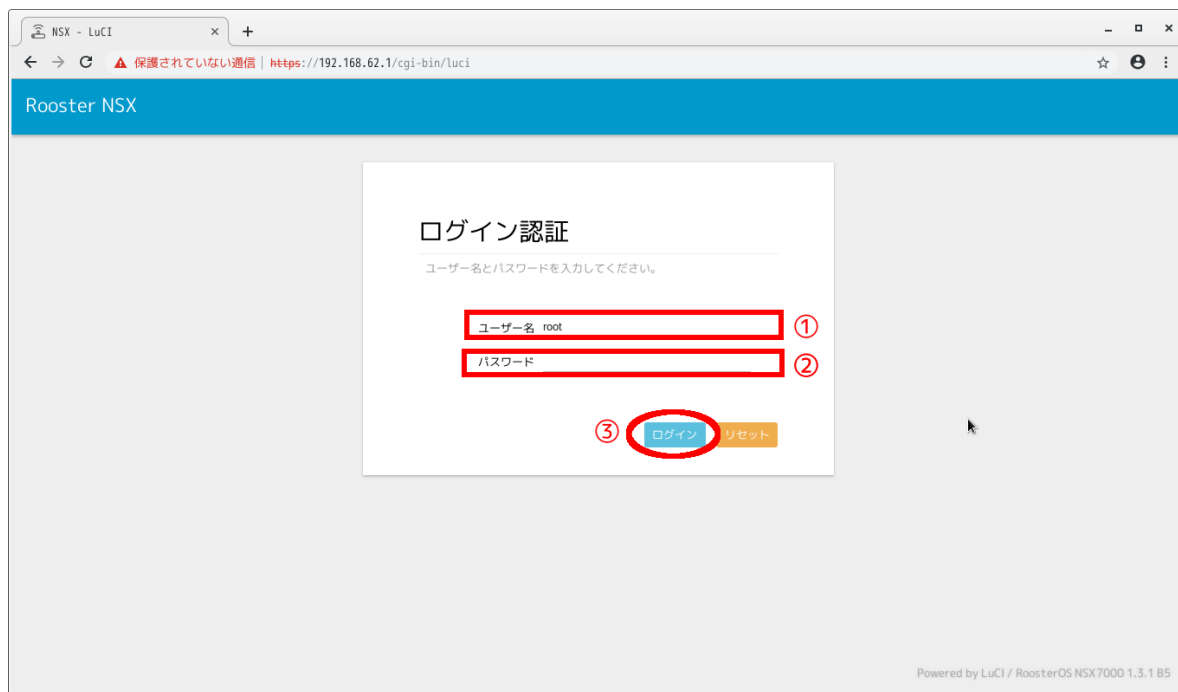
6.2 NSX7000 の Web UI を使って SSH ログイン設定を行なう

Google Chrome ブラウザを起動して NSX7000 にアクセスし, SSH ログイン設定を行います. 以下の説明では IP アドレス 192.168.62.1 が NSX7000 に割り当てられているものとしています.

1. Google Chrome ブラウザを起動して NSX7000(IP アドレス: 192.168.62.1) にアクセスする

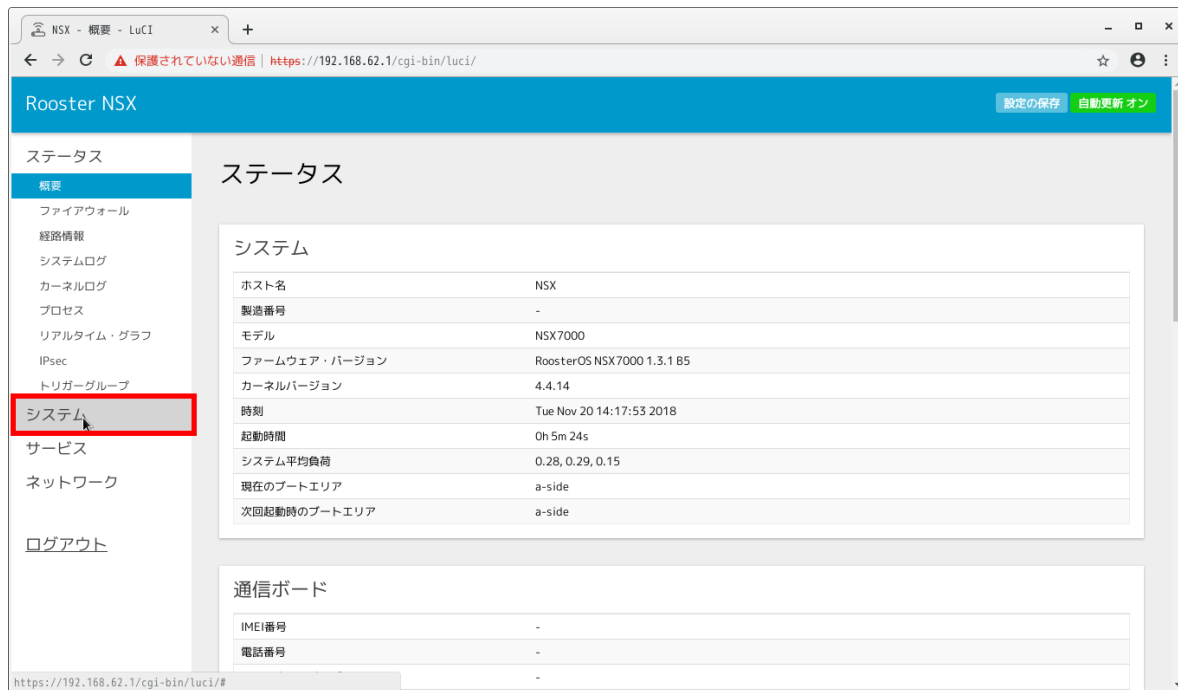


2. 「ログイン認証」画面が表示されたら、ユーザ名に「root」とそのパスワードを入力し、「ログイン」ボタンをクリックする

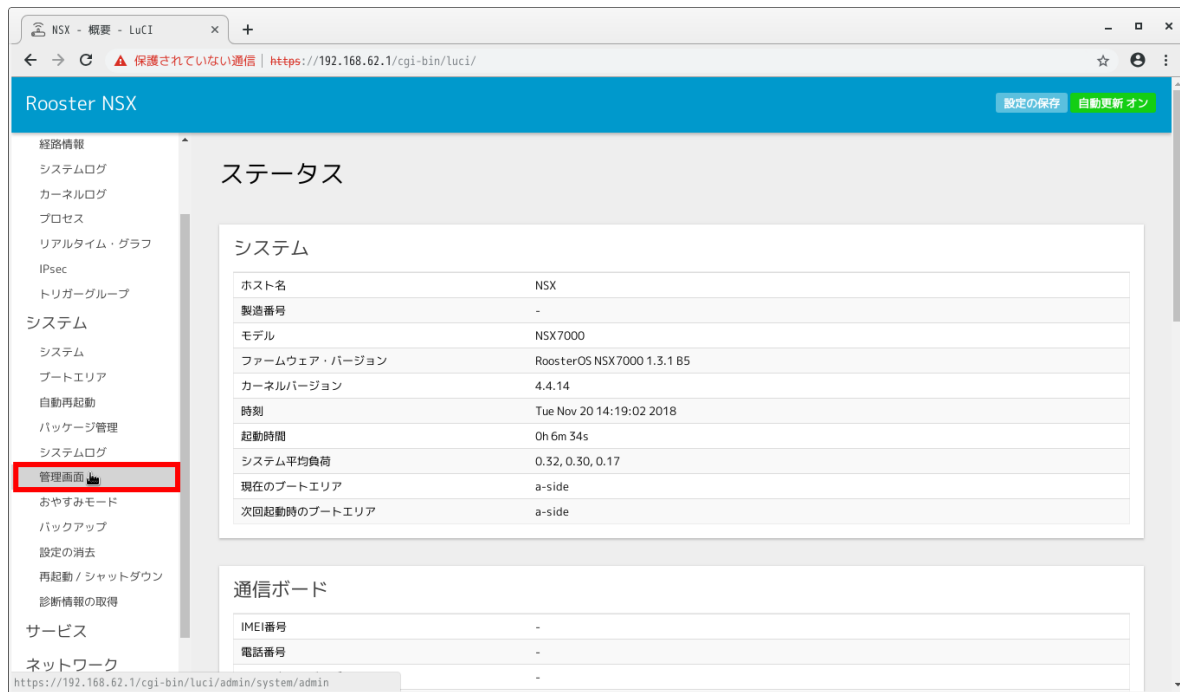


注釈: 工場出荷時の root アカウントのパスワードは root です

3. 「ステータス」画面が表示されたら、「システム」メニューをクリックしてサブメニューを表示する



4. 「管理画面」サブメニューをクリックする

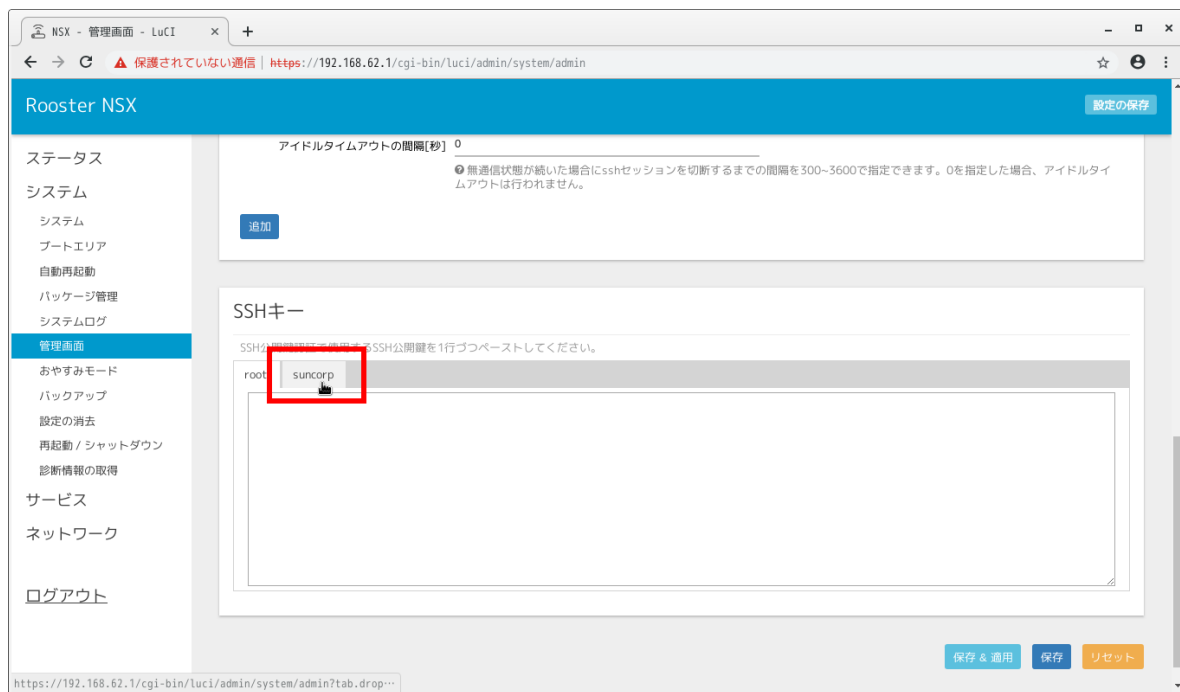


5. 「管理画面」画面が表示されたら「SSH アクセス」エリアでSSH 設定を行なう。

- 「SSH サーバを有効にする」: チェックを入れ, 有効にする
- 「インタフェース」: (必要ならアクセス可能なインタフェースを指定する)
- 「ポート」: (必要なら SSH サーバが待ち受けるポート番号を変更する)
- 「パスワード認証」: チェックをはずし, パスワード認証を許可しない
- 「パスワードを使用した root 権限でのログインを許可する」: チェックを外し, ログインを許可しない
- 「ゲートウェイ・ポート」: チェックをはずす
- 「キープアライブの間隔」: (必要なら指定する)
- 「アイドルタイムアウトの間隔」: (必要なら指定する)



6. 「管理画面」画面を下にスクロールし、「SSH キー」を表示する。suncorp タブをクリックする

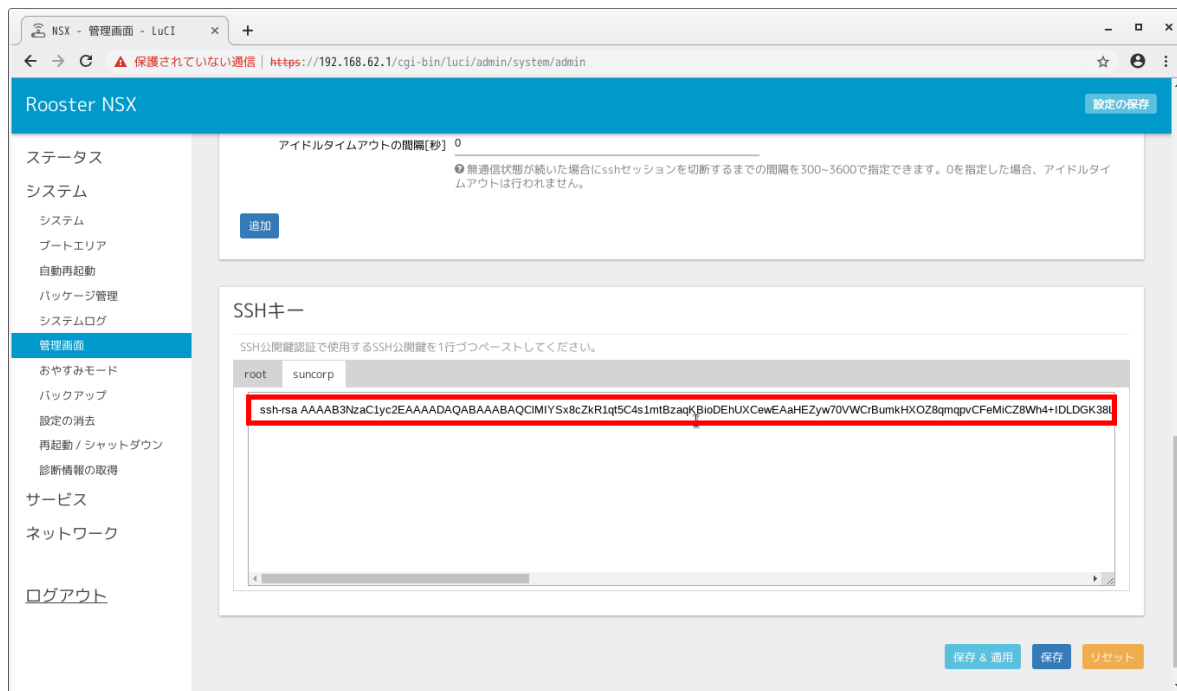


7. cat コマンドで\$HOME/.ssh/id_rsa.pub を端末上に表示する

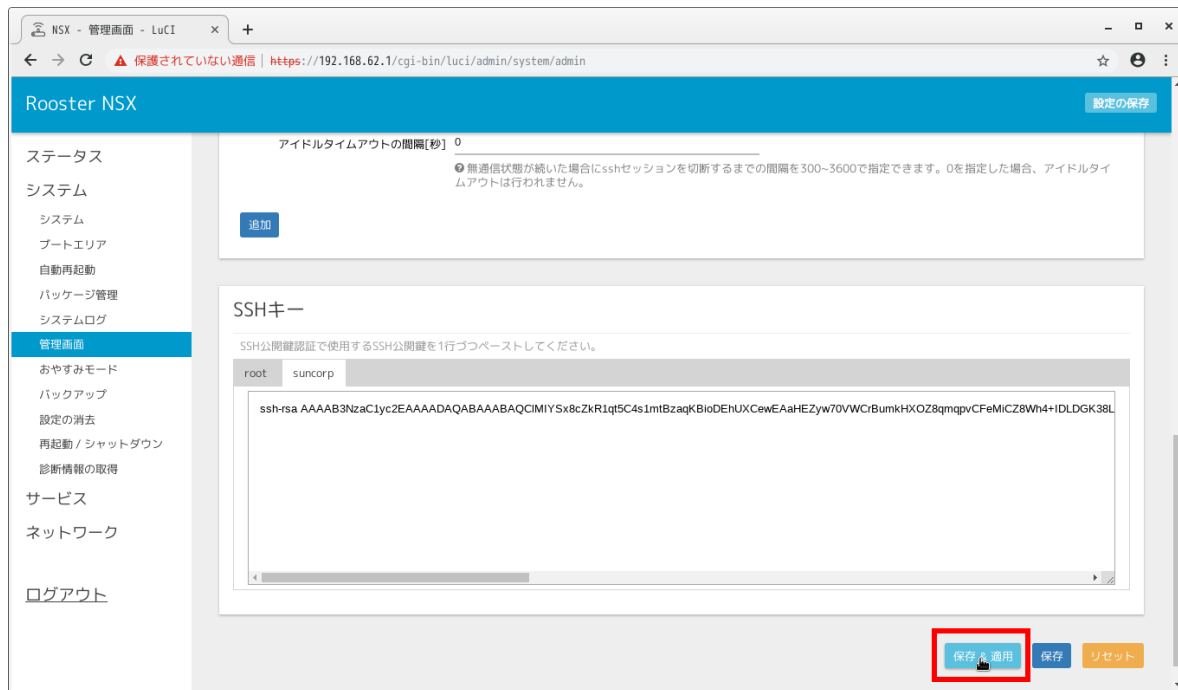
```
$ cat $HOME/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACoutrFr1wQO8H8jWWDKOk+dxa+
/qDAbgjq97WtWeMAI8w4JJZuVXW+TRaPKcDvhXxdZGLZNNB0Y9xBMrG2V7ZqNpSV
O3fTPhm/n5v7nqtCRvWXv7euIr6oXxHuHAC6EFjx+nlhz9FzRuFKwj0IUhM5p5JI
TImALEBj+lexjyKRrjXoGraXuvYNQ8mN8/05gdnwDfoY90K8S54/GEMJeA9lBxNo
TjCbvhNb+ge51kzrDwcX6QLQY5IKXBv9NRSWjdh82hFST94dVlXkPgafTOD9EmV4
bKfLZDcjMAidRdOSpEcPV/+WvfTGJX4O3OE4zQwgjbQsDaglOwfw3IhzFErUBaYC
lnNW94p4n1Q3CwS/CeWQVrntE99ig+dfppMszlOCGep8fsBe19ryX29L8fZHUw/G
L/66iXD3lbdkoQgUMPwsJ4IJOpr1N22zDuHSsYn1N8DYabrZM/bY2bwHTHHdW2oK
X08+Pf5g+ZMudoMxZdVIHOCUFUkf5gCnVoulbXxulWBO2yWHGnfzT3yqjzHUDXQ
g0DFLlGIhoMUQAoNX4VuC83nsaZSstorZBI503aYnVdTecv3RHZ4U/MXsKG53MMv
k/r5jchXChQZsYN5isE/ThzySFIYRn8GJIw0Xgx99hALdtsI5ZShl7igKU1HH6yk
C1LLqpTb9BB/tQz/7w== m2m@debian
```

8. マウスを使って前述の「cat \$HOME/.ssh/id_rsa.pub」の出力をコピーする

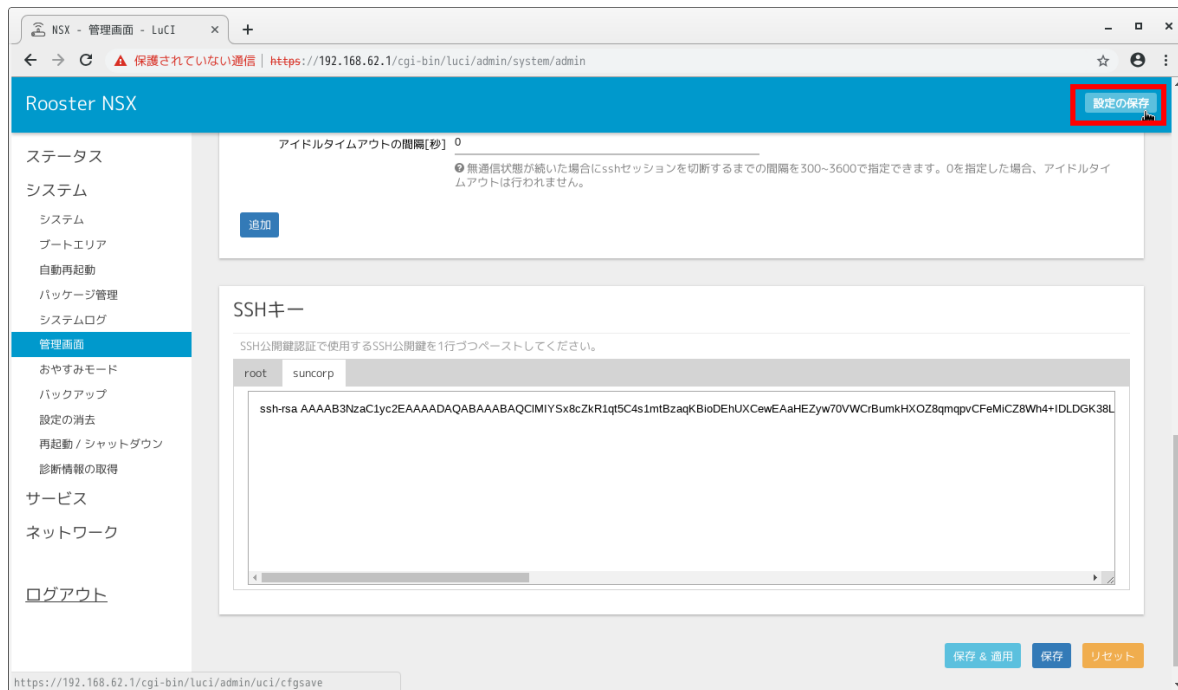
9. suncorp タブに SSH ログイン用の公開鍵ファイルの内容をペーストする



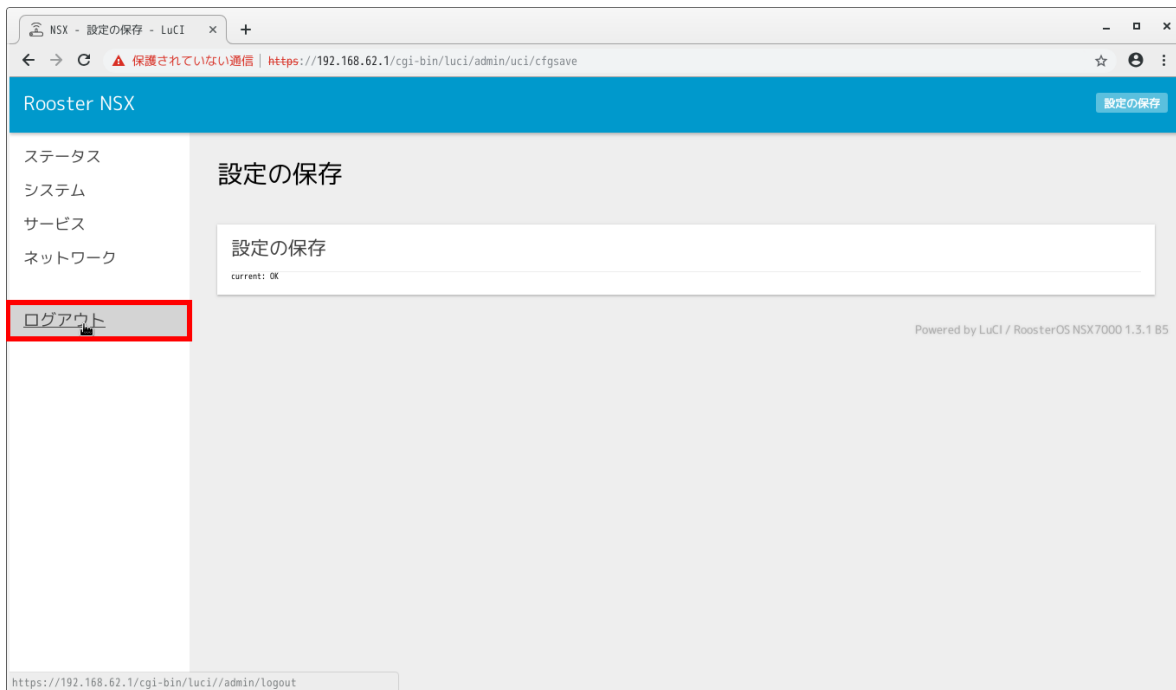
10. 「保存&適用」ボタンをクリックし、変更した設定を反映する



11. 「設定の保存」ボタンをクリックし、NSX7000 のフラッシュメモリに設定を書き込む



12. 設定の保存が終了したら「ログアウト」メニューをクリックしてログアウトする



13. Google Chrome ブラウザを閉じる

6.3 SSH ログインできることを確認する

ssh コマンドを使って SSH ログインできることを確認します。

1. (オプション) 毎回のパスフレーズ入力を避けるために ssh-agent を起動し、パスフレーズを登録する

```
$ [ -n "$SSH_AUTH_SOCK" ] || eval $(ssh-agent -s)
$ ssh-add $HOME/.ssh/id_rsa
Enter passphrase for /home/m2m/.ssh/id_rsa: <パスフレーズを入力>
Identity added: /home/m2m/.ssh/id_rsa (/home/m2m/.ssh/id_rsa)
```

2. ssh コマンドを使い、**suncorp** アカウントで NSX7000(IP アドレス: 192.168.62.1) に SSH ログインする

```
$ ssh suncorp@192.168.62.1
```

SSH ログインの成功が確認できたら、NSX7000 上で exit コマンドを実行してログアウトしてください。

第 7 章

NSX7000 の開発者モードを ON にする

`rooster-os-developer-mode` コマンドを使うことで下記を行えます。

- 開発者モードの ON/OFF を表示する
- 次回起動時に開発者モードを ON になるように設定する
- 次回起動時に開発者モードを OFF になるように設定する

NSX7000 に SSH ログインし, 次回起動時に開発者モードを ON になるように設定して, NSX7000 を再起動してください. 再起動後, NSX7000 の開発者モードが ON になります.

7.1 開発者モードの ON/OFF を表示する

NSX7000 に SSH ログインし, root アカウントに変更します. root アカウントへの変更には `su` コマンドを使用します. `su` コマンド実行時に尋ねられるパスワードは Web UI のログイン認証で使用するパスワードです.

```
$ ssh suncorp@192.168.62.1
BusyBox v1.24.2 () built-in shell (ash)

      _____
     /  _  \  _  /  _  /  _  /  _  \  _  /
    /  /  /  _  \  _  \  _  /  _  \  _  /  /  \  \
   /  _  /  /  /  /  /  (  _  )  /  /  _  /  /  /  /  /
  /  /  |  |  \  _  /  _  \  _  /  _  \  _  /  /  /  /

suncorp@NSX:~$ su
Password:

BusyBox v1.24.2 () built-in shell (ash)

root@NSX:/home/suncorp#
```

rooster-os-developer-mode コマンドの **show** サブコマンドで現在の開発者モードの ON/OFF と次回起動時の開発者モードの ON/OFF を確認できます。

```
root@NSX:/home/suncorp# /safe/bin/rooster-os-developer-mode show
current: off
next    : off
```

「current」行に表示されているのが現在の開発者モードの状態です。「on」なら開発者モードは ON です。「off」なら開発者モードは OFF です。

「next」行に表示されているのが次回起動時の開発者モードの状態です。「on」なら次回起動時の開発者モードは ON です。「off」なら次回起動時の開発者モードは OFF です。

7.2 次回起動時に開発者モードを ON になるように設定する

rooster-os-developer-mode コマンドの **on** サブコマンドで次回起動時の開発者モードを ON に設定できます。設定の変更には約 30 秒かかります。

```
root@NSX:/home/suncorp# /safe/bin/rooster-os-developer-mode on
```

7.3 次回起動時に開発者モードを OFF になるように設定する

rooster-os-developer-mode コマンドの **off** サブコマンドで次回起動時の開発者モードを OFF に設定できます。設定の変更には約 30 秒かかります。

```
root@NSX:/home/suncorp# /safe/bin/rooster-os-developer-mode off
```

7.4 次回起動時の開発者モードを ON になるように設定して再起動する

rooster-os-developer-mode コマンドで次回起動時の開発者モードを ON に設定し、NSX7000 を再起動してください。

```
root@NSX:/home/suncorp# /safe/bin/rooster-os-developer-mode on
root@NSX:/home/suncorp# /safe/bin/rooster-os-developer-mode show
current: off
next    : on
root@NSX:/home/suncorp# reboot
```

第 8 章

おわりに

環境構築は以上で終了です。

Add-on アプリケーションの開発の進め方については別文書「Add-on アプリケーション開発-パッケージ作成マニュアル-」を参照してください。

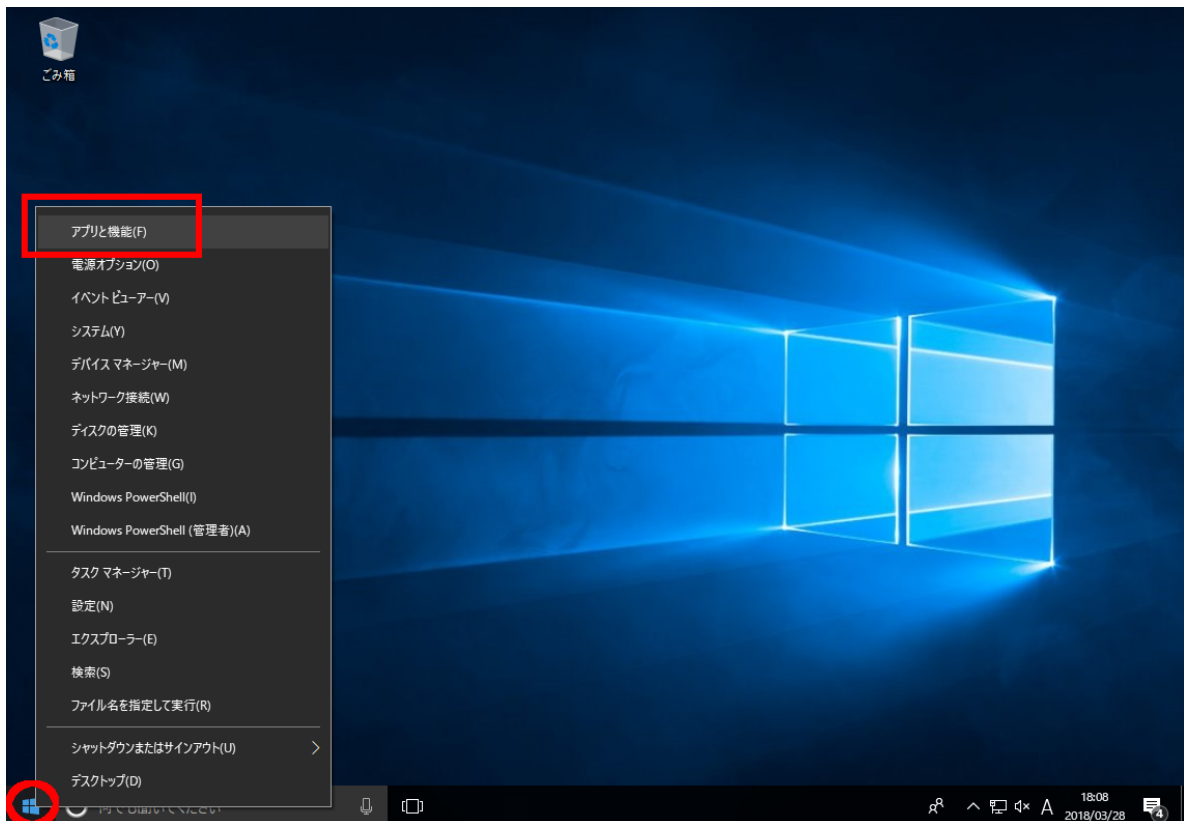
付録 A

Ubuntu を Windows 10 Pro にインストールする

Ubuntu を Windows10 Pro にインストールする手順を説明します。

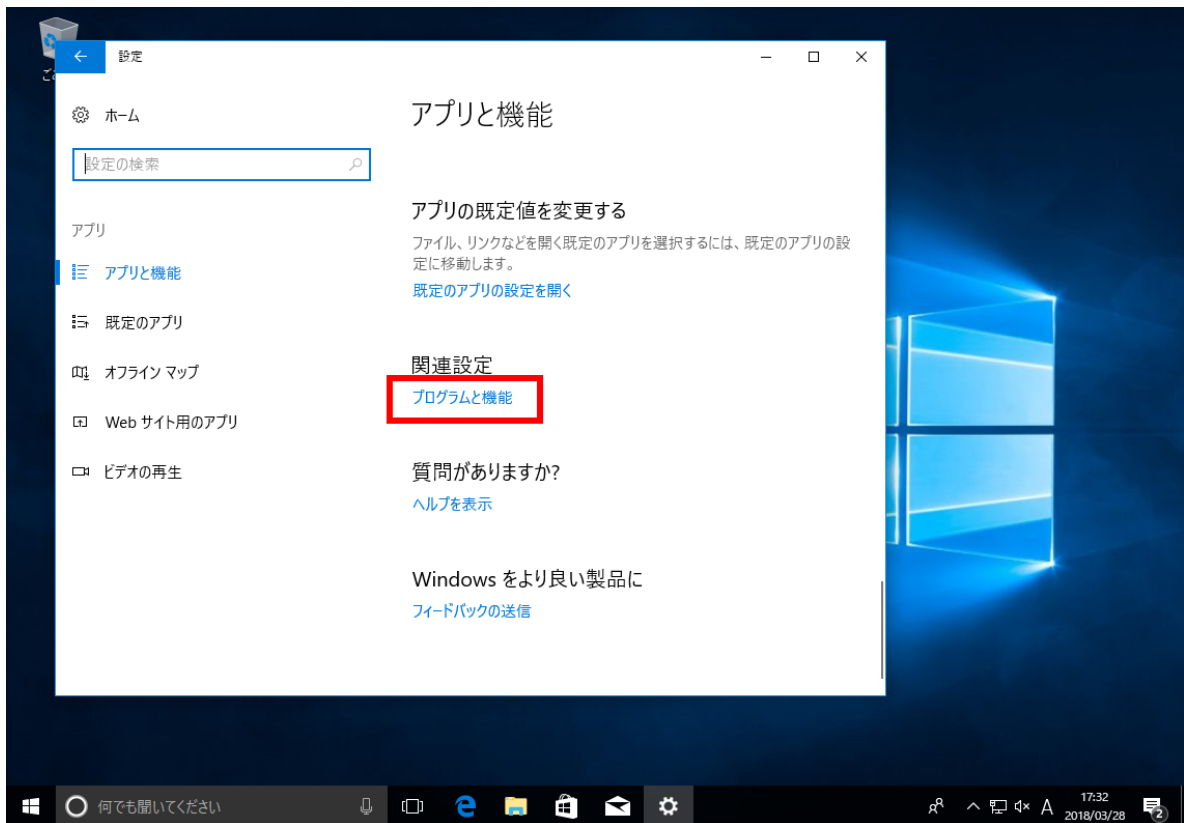
32 ビット版 Windows10 Pro にはインストールできません。また、Windows10 Home は本文書の対象外^{*1}です。

1. スタートボタンを右クリックし、メニューから「アプリと機能」をクリックします。

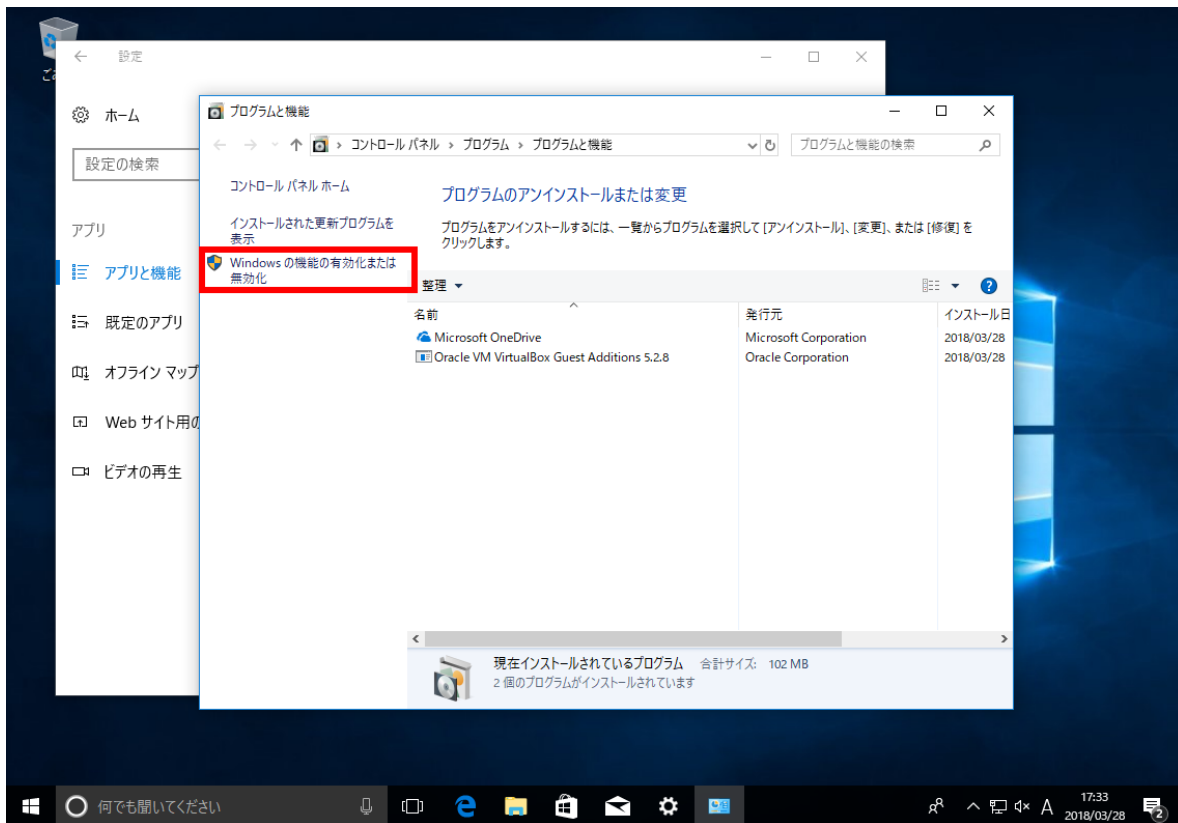


^{*1} 未検証です。

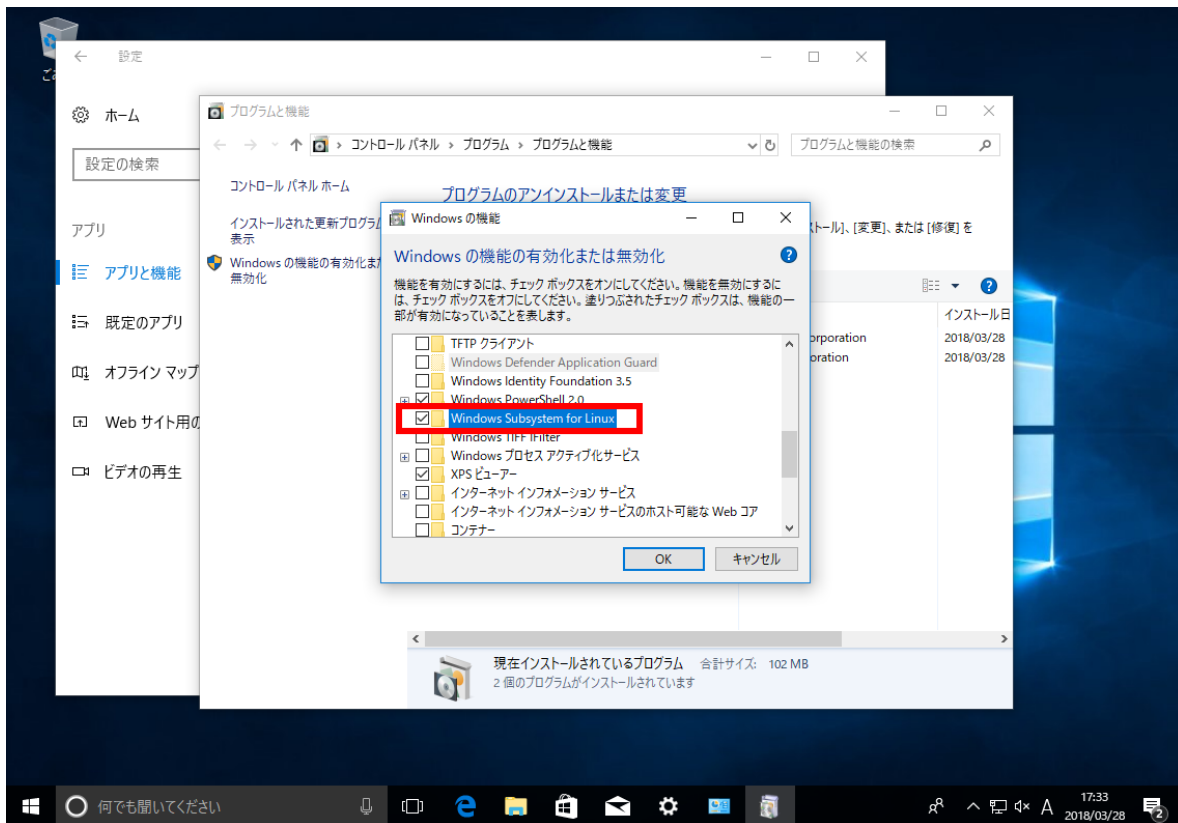
2. 「プログラムと機能」をクリックします。



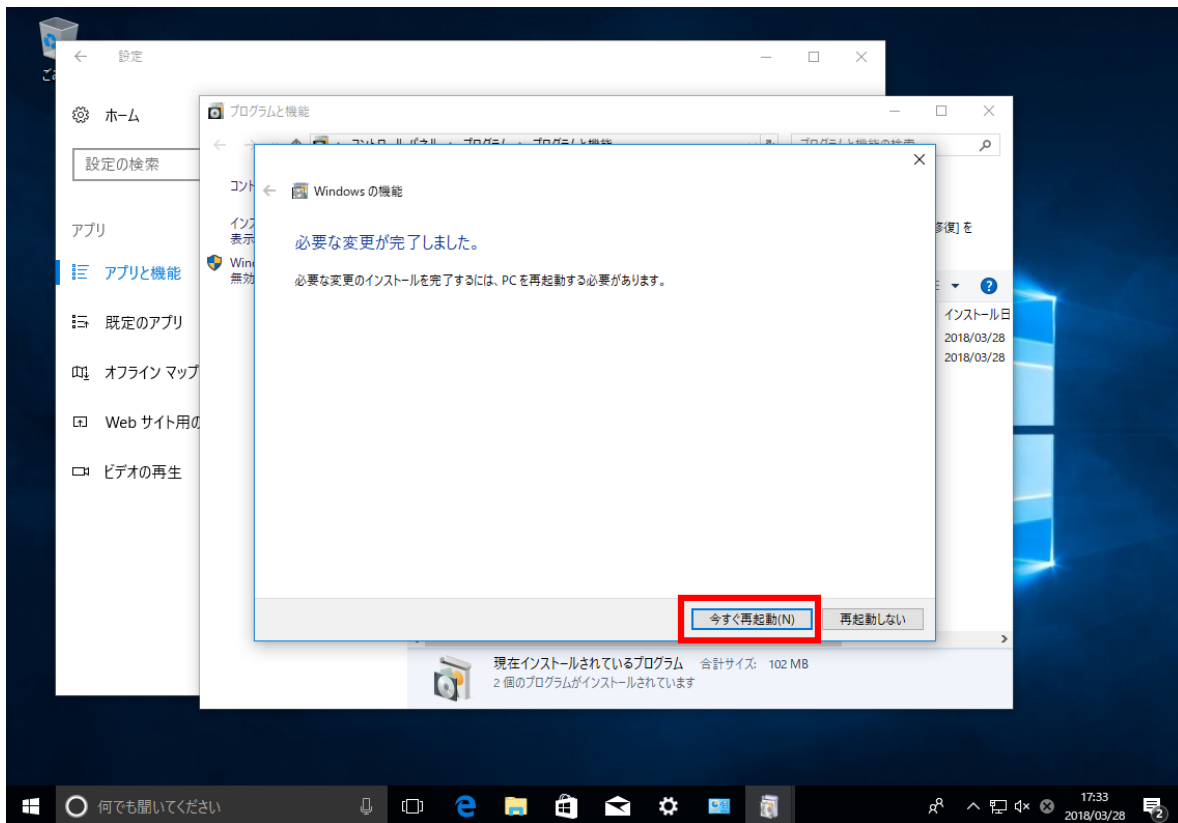
3. 「Windows の機能の有効化または無効化」をクリックします。



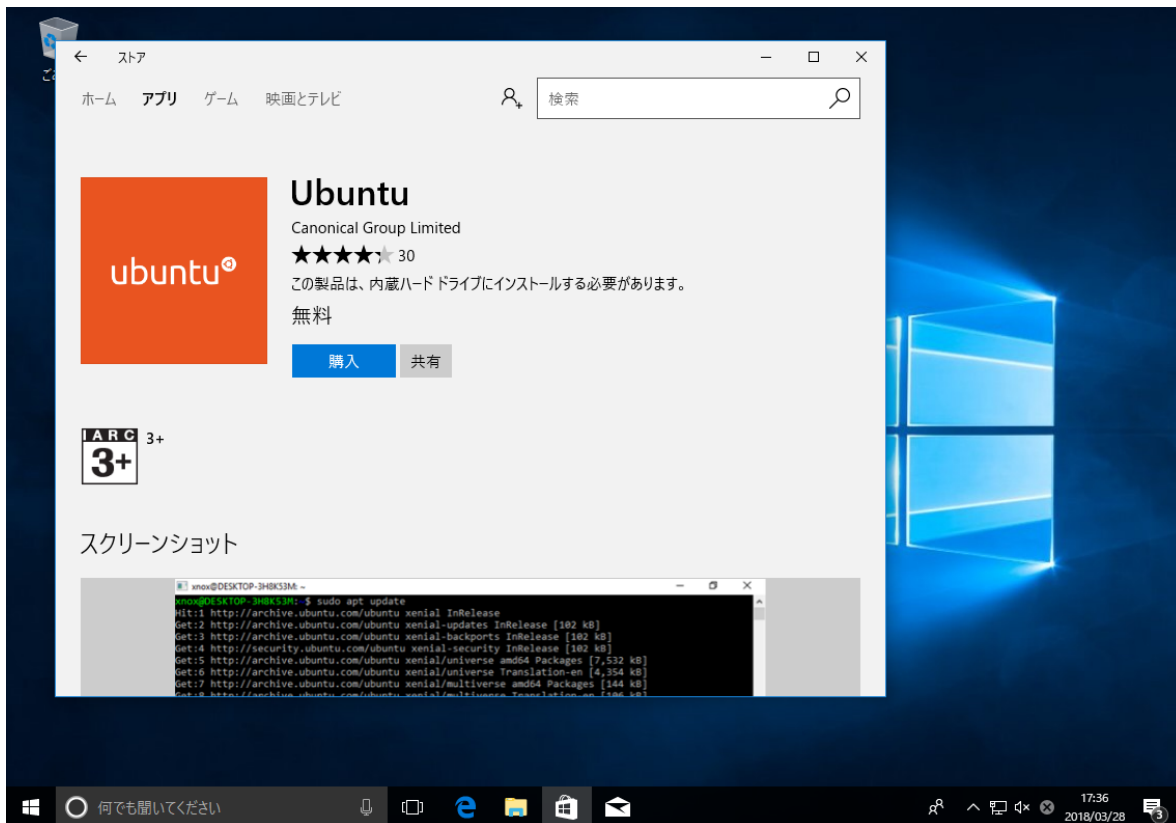
4. 「 Windows Subsystem for Linux 」にチェックを入れ、「 OK 」ボタンをクリックします。



5. 「今すぐ再起動」ボタンをクリックします。



6. Windows 再起動後, Windows ストアから Ubuntu をダウンロードし, インストールします.



付録 B

開発者モード OFF 時の NSX7000 での制限

機密保持, および, ルータ設定の不整合を防ぐため, root 権限を持つプロセスであっても一部の機能が制限されます. 開発者モードが OFF のとき, root 権限を持つアプリケーションでも下記のカーナビリティを必要とする機能は利用できません.

- CAP_MAC_ADMIN
- CAP_MAC_OVERRIDE
- CAP_SETFCAP
- CAP_SYS_PTRACE
- CAP_SYS_RAWIO
- CAP_AUDIT_CONTROL
- CAP_AUDIT_READ
- CAP_AUDIT_WRITE
- CAP_BLOCK_SUSPEND
- CAP_FSETID
- CAP_LINUX_IMMUTABLE
- CAP_MKNOD
- CAP_NET_ADMIN
- CAP_SYS_ADMIN
- CAP_SYS_MODULE
- CAP_SYS_NICE
- CAP_SYS_RESOURCE

- CAP_SYSLOG
- CAP_WAKE_ALARM

B.1 開発者モード OFF 時に root 権限を持ったアプリケーションでできないことの例

- mount コマンドや mount システムコールを使った USB メモリのマウント処理^{*1}
 - CAP_SYS_ADMIN ケーパビリティを持たないため
- ip コマンドや iptables コマンド, route コマンドによるネットワーク設定の変更処理^{*2}
 - CAP_NET_ADMIN ケーパビリティを持たないため
- reboot システムコールを使ったシャットダウンおよびリブート処理^{*3}
 - CAP_SYS_ADMIN ケーパビリティを持たないため

^{*1} suncorp アカウントで SSH ログインし su コマンドで root アカウントに変更した後 (もしくは root アカウントで SSH ログインした後), 手動で mount コマンドを実行して USB メモリをマウントすることはできません

^{*2} suncorp アカウントで SSH ログインし su コマンドで root アカウントに変更した後 (もしくは root アカウントで SSH ログインした後), 手動でネットワーク系コマンドを実行してネットワーク設定を変更することはできません

^{*3} suncorp アカウントで SSH ログインし su コマンドで root アカウントに変更した後 (もしくは root アカウントで SSH ログインした後), poweroff コマンドや reboot コマンドによるシャットダウンおよびリブート処理はできません

付録 C

開発者モード ON 時の NSX7000 での制限

開発者モードが ON のとき, `rooster-os-cap-add-on` コマンドを使用することで任意のカーナビリティを有効にすることができます。しかし, 開発者モードが ON であっても, 下記のカーナビリティを有効にすることはできません。

- `CAP_MAC_ADMIN`
- `CAP_MAC_OVERRIDE`
- `CAP_SETFCAP`
- `CAP_SYS_RAWIO`