Orchestrating a brighter world

NEC

HaaS コントローラ
導入マニュアル

改訂履歴

版数	改訂日	改訂内容
0.1	2020/9/30	ドラフト版
0.2	2020/10/30	0.1 版に以下の修正を行いました。 ・書式・表記ゆれの統一、説明の改善（全体） ・旧 2.4　パーティションの設定　を削除 ・2.4（旧 2.5）　Zabbix Agent の設定　の TBD としていた手順を追加 ・5.1　ログ収集設定　の TBD としていた手順を追加
0.3	2020/11/13	0.2 版に以下の修正を行いました。 ・全体的に、説明の改善、誤記修正 ・1.1.1　システム構成図　および　1.2.1　ネットワーク構成図　の図を 3 冗長想定に変更 ・2.4　Zabbix Agent の設定　に SELinux 無効化とファイアウォール設定の手順を追加
0.4	2020/12/03	0.3 版に以下の修正を行いました。 ・6.4.9　Zabbix Agent バージョン確認　の入力フォーマットを修正
1.0	2020/12/11	0.4 版に以下の修正を行いました。 ・全体的に書式・表記ゆれの統一、説明の改善、誤記修正 ・2.4　Zabbix Agent の設定　にバージョン確認の手順を追加 ・3.2　HaaS VM イメージ配備　の手順修正 ・旧 6.2　インストールパッケージ確認　を削除 ・3.4　After config 機能　を追加
2.0	2021/3/15	1.0 版に以下の修正を行いました。 ・全体的に書式・表記ゆれの統一、説明の改善、誤記修正 ・2.1　HaaS ホストの環境構築　の手順から公開鍵作成手順を削除、SNMP 設定手順および cron 設定手順を追加 ・2.4　Zabbix Agent の設定　の config ファイル修正箇所と内容の変更 ・2.5　wheel グループユーザの sudo 許可設定　を追加 ・2.6　ユーザ作成　を追加 ・2.7　ssh での root ログイン禁止設定　を追加 ・3.2　HaaS VM イメージ配備　の XML 修正手順に MAC アドレス修正を追加 ・3.4　After config 機能　の手順修正 ・3.5　ユーザ作成　を追加 ・3.6　証明書関連設定　を追加 ・6.1.1　PM OS バージョン／カーネルバージョン情報の確認　および 6.1.2　VM OS バージョン／カーネルバージョン情報の確認　のカーネルバージョン更新 ・6.2.1　PM IP アドレス／サブネットマスクの確認　および 6.2.2 VM IP アドレス／サブネットマスクの確認の設定確認内容修正
3.0	2021/3/23	版数を 3.0 に修正（他マニュアルとの版数統一のため）
3.4	2021/6/18	3.0 版に以下の修正を行いました。 ・全体的に書式・表記ゆれの統一、説明の改善、誤記修正 ・2.1　HaaS ホストの環境構築　の snmp 起動手順を削除 ・4 ポート化に伴い内部/外部通信用ネットワーク設定手順を修正（2.2 章、2.3 章） ・2.4　Zabbix Agent の設定　に zabbix-agent の自動起動設定手順を追加 ・2.5　NTP 設定　を追加 ・3.2　HaaS VM イメージ配備　に VM イメージ非圧縮化の手順を追加 ・3.4　After config 機能　に OAuth2 認証情報登録手順を追加 ・3.5　cron 設定　を追加（2.1 章に記載していた手順を移動）

版数	改訂日	改訂内容
4.0	2021/6/30	3.4 版に以下の修正を行いました。 ・コメントアウトの方法を明記 ・3　HaaS VM 構築　の VM イメージ、XML 格納場所を明記 ・3.5　cron 設定　に VM 名確認の手順を追加 ・3.4　After config 機能　の STEP5-4、5-5 について記載を修正 ・3.4　After config 機能　の STEP7-3 についてラック名の説明を修正
4.2	2021/8/23	4.0 版に以下の修正を行いました。 ・2.3　外部通信用ネットワークの設定　にデフォルトゲートウェイ設定手順を追加 ・3　HaaS VM 構築　に HaaS 構築時に発生する障害情報を追記 ・3.7　証明書関連設定　の手順を見直し ・6.3　ミドルウェアバージョン確認　に docker コンテナから抜ける手順を追加
4.3	2021/9/13	4.2 版に以下の修正を行いました。 ・2　PM の設定　の冒頭に、全ての HaaS PM で実施する旨を追記 ・3　HaaS VM 構築　の qcow2、xml ファイル名の例を修正 ・3.2　HaaS VM イメージ配備　の xml 変更手順を修正 ・3.7　証明書関連設定　の/etc/hosts 設定手順を別紙に移動
4.4	2021/11/8	4.3 版に以下の修正を行いました。 ・2.3　外部通信用ネットワークの設定　の VLAN ID の記載を変数化 ・2.4　Zabbix Agent の設定　の設定ファイル修正箇所を修正 ・3.4　After config 機能　の STEP5-9 について変数説明を修正 ・7.3　ユーザパスワード変更　を削除
4.5	2021/11/25	4.4 版に以下の修正を行いました。 ・2.2　内部通信用ネットワークの設定　に手順を追加 ・3.7　証明書関連設定　の手順を修正 ・4　CMDB 設定　を削除
5.0	2021/12/27	4.5 版に以下の修正を行いました。 ・3.4　After config 機能　の一部変数説明の修正 ・3.7　証明書関連設定　の一部記載修正
5.1	2022/02/10	5.0 版に以下の修正を行いました。 ・全体　sudo su - コマンドの sudo を削除 ・全体　「注： root 権限を使った操作をする場合」→「注： root 権限を使った操作をする為」に修正。 ・全体　コンソールログインの手順に”「注： root 権限を使った操作をする為」～”の注意書きを追記 ・はじめに　第 4 章 CMDB 設定の記述を削除 ・2.6　wheel グループユーザの sudo 許可設定　を削除 ・3.4　After config 機能　の STEP2-2、STEP3-2 の参考値及び変数説明を修正 ・3.4　After config 機能　のホスト情報設定に objectClass を追加 ・1.3.2　ソフトウェア要件　の pacemaker のバージョンを更新 ・5.1.2　VM OS バージョン／カーネルバージョン情報の確認　のカーネルバージョンを更新 ・5.2.3　PM NTP の確認　の「*」、「-」の記述追加 ・5.2.4　VM NTP の確認　の「*」、「-」の記述追加 ・5.3.1　Pacemaker バージョン確認　のバージョン情報を更新

版数	改訂日	改訂内容
5.2	2022/02/28	5.1 版に以下の修正を行いました。 ・2.3　外部通信用ネットワークの設定　STEP2 にて「2-1」→「2-2」に誤記修正 ・3.4　After config 機能　STEP3 にて「3-1～3-7」→「3-1～3-6」に誤記修正 ・3.7　証明書関連設定　STEP11-2 にて「STEP10-1 で」→「STEP11-1 で」に誤記修正 ・3.7　証明書関連設定　STEP11-4 にて「STEP10-3 で」→「STEP11-3 で」に誤記修正 ・3.7　証明書関連設定　■NE-OPS→HaaS の ssh/sftp 通信設定　にて「(STEP1～STEP6)」→「(STEP1～STEP5)」に誤記修正 ・5.3　ミドルウェアバージョン確認　にて「本稿の 6.3.1～6.3.13」→「本稿の 5.3.1～5.3.13」に誤記修正
5.3	2022/04/25	5.2 版に以下の修正を行いました。 ・全体　XML ファイルの格納先を“/etc/libvirt/qemu”→“/tmp”に修正 ・1.4.2 ワークフロー　にて sudo 許可設定を削除 ・3.2　HaaS VM イメージ配備　STEP7 にて VM イメージのファイルパスを修正 ・3.4　After config 機能　にて「01_interface_ip_address_change_inventory.yml」の表示イメージの修正 ・3.4　After config 機能　にて　/02_construction_change.sh 実行時の注意事項を追記 ・3.4　After config 機能　STEP1-1、2-1　にてコンソールログイン時の Enter 入力手順を追加 ・3.4　After config 機能　STEP2-4、3-4　にて VM 起動確認手順を追加 ・3.4　After config 機能　の STEP3-2 にて「内部 NTP」→「外部 NTP」に修正 ・3.7　証明書関連設定　証明書の導通確認にてエラー時の対応内容を追記 ・3.8　事後作業　を追加（VM 構築後の不要なファイルの削除）
5.4	2022/11/30	5.3 版に以下の修正を行いました。 ・1.3.1　ネットワーク要件　図 1.5　外部インタフェースに DHCP を追加 ・2.4　Zabbix Agent の設定　「nul」→「null」の誤記修正

はじめに

■このマニュアルについて

このマニュアルでは、HaaS コントローラの導入にあたり必要な設定や基本的な操作について記載します。

■マニュアルの前提条件

このマニュアルの手順を実施した際に、エラーまたは想定以外の状態になった場合は、障害原因によるエラー等も考えられるため、障害対応に移行して対応をお願いします。併せて、システム管理者へもお問い合わせください。

■マニュアルの構成について

このマニュアルは、次の章で構成されます。

章番号	記述内容
第 1 章　概要	HaaS コントローラのシステムやネットワークの概要、および導入作業の前提条件について説明しています。
第 2 章　PM の設定	HaaS ホストの環境構築、ネットワークや Zabbix Agent の設定について説明しています。
第 3 章　HaaS VM 構築	HaaS VM のイメージファイル配備、HaaS VM の起動、After config 機能および証明書関連の設定手順について説明しています。
第 4 章　各種設定	ログ収集の設定について説明しています。
第 5 章　インストールファイルとサーバ設定の確認	インストールファイルや PM・VM サーバ設定の確認手順、ミドルウェアのバージョン確認手順について説明しています。
第 6 章　システム動作の確認	クラスタ状態の正常性確認や外部装置との疎通確認、ユーザパスワード変更の手順について説明しています。

■関連するマニュアル

- 『BoxUP 環境定義書』
- 『BoxUP 環境定義書_IP アドレス』
- 『BoxUP 環境定義書_ホスト名』
- 『HaaS コントローラ操作マニュアル』

■マニュアルの表記について

このマニュアルでは、次のような表記を使用しています。

マニュアル表記例	説明
注：	操作する上で必ず守らなければならない注意事項や制限事項、および誤りやすい操作について説明しています。
参考：	知っておくと役に立つ情報、および操作のアドバイスなどについて説明しています。
「3.1 サービス影響」	マニュアル内の参照先を「 」で囲んで表記しています。
『HaaS コントローラ操作マニュアル』	参照する他のマニュアルを『 』で囲んで表記しています。
［OK］ボタン	メニュー名、画面名、ボタン名、フィールド名などを［ ］で囲んで表記しています。
「ログ」を選択	選択項目、表示内容などを「 」で囲んで表記しています。
＜イメージファイル名＞	コマンド入力フォーマットの説明では、変数を＜ ＞で囲んで表記しています。

■著作権について

■商標について

目次

1 概要

HaaS コントローラは、ライフサイクル制御機能、構成管理機能、障害監視機能などの、ハードウェアの管理機能を有する装置です。

1.1 システム構成

1.1.1 システム構成図

HaaS コントローラのシステム構成図を以下に示します。

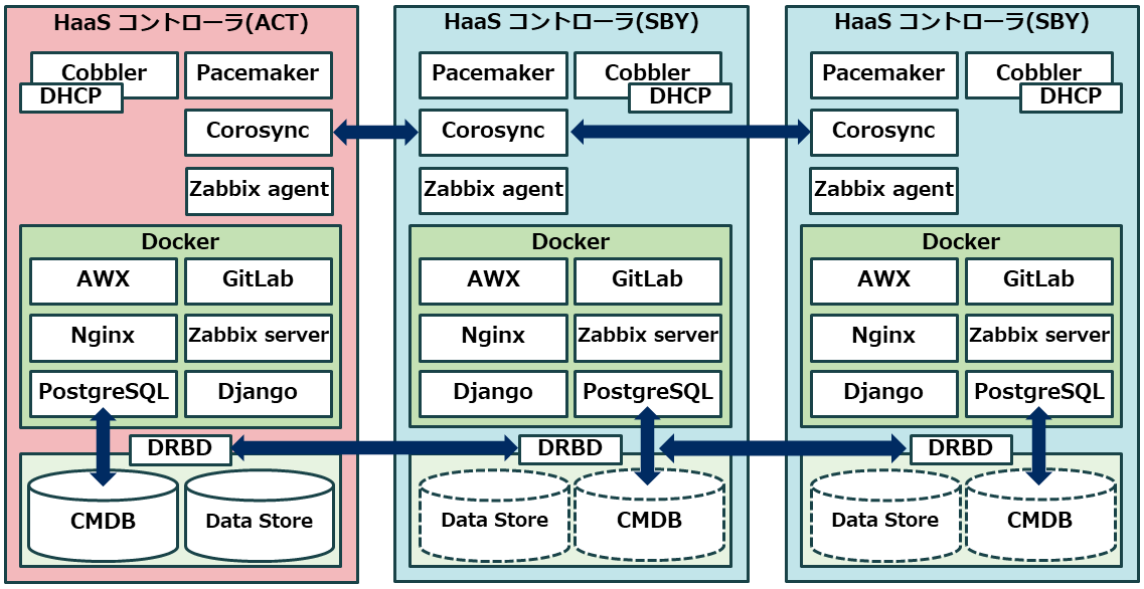

図 1-1 システム構成図

1.1.1.1 冗長構成

「図 1-1 システム構成図」に示すとおり、Pacemaker および Corosync によるクラスタ制御および DRBD によるストレージレプリケーションによって、HA クラスタ構成を実現し、可用性を確保しています。

1.1.2　ハードウェア構成

HaaS コントローラのハードウェア外観図を以下に示します。

図 1-2　ハードウェア外観図（Dell PowerEdge R640 の場合）

参考：　ハードウェア要件については、「1.3.2　ハードウェア要件」を参照してください。

1.1.3　ソフトウェア構成

HaaS コントローラのソフトウェア構成を以下に示します。

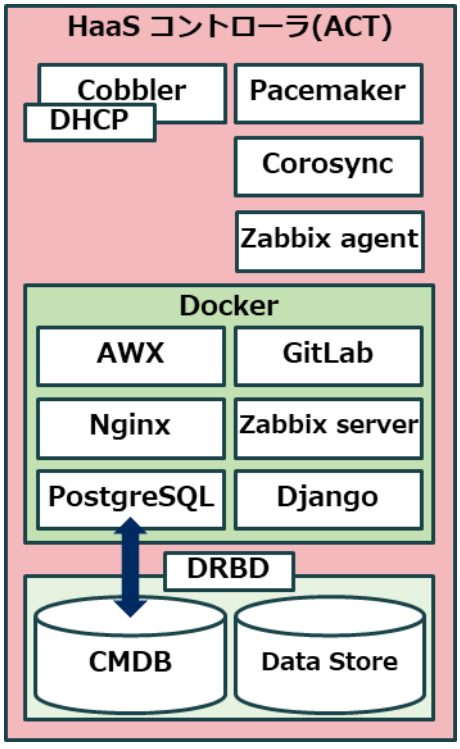

図 1-3　ソフトウェア構成図

参考：　ソフトウェア要件については、「1.3.3　ソフトウェア要件」を参照してください。

1.2 ネットワーク構成

1.2.1 ネットワーク構成図

HaaS コントローラのネットワーク構成図を以下に示します。

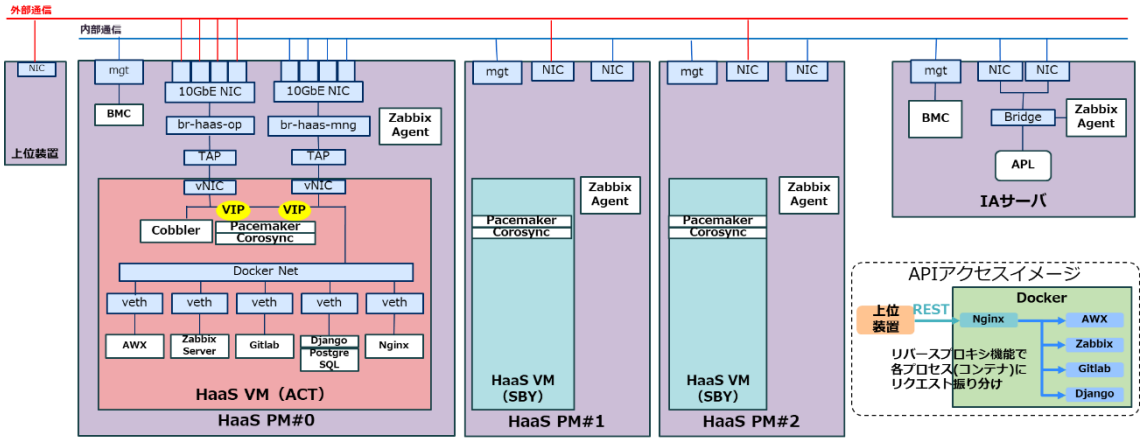

図 1-4 ネットワーク構成図

1.2.2 物理ネットワーク構成

参考： HaaS コントローラを含めた物理ネットワークについては、『BoxUP 保守マニュアル』の「1.3.1.2 物理ネットワーク」を参照してください。

1.2.3 論理ネットワーク構成

参考： HaaS コントローラを含めた論理ネットワークについては、『BoxUP 保守マニュアル』の「1.3.1.3 論理ネットワーク」を参照してください。

1.3 システム要件

1.3.1 ネットワーク要件

HaaS コントローラの外部インタフェースを以下に示します。

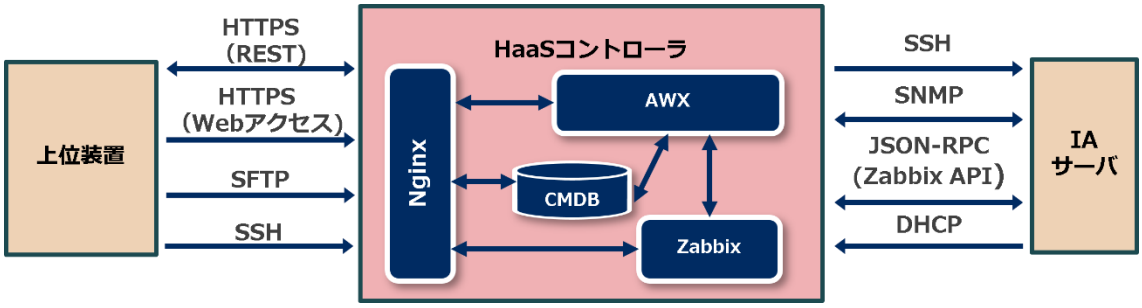

図 1-5 外部インタフェース

1.3.2 ハードウェア要件

ハードウェア諸元を以下に示します。

表 1-1 ハードウェア諸元（Dell PowerEdge R640 の場合）

インタフェース	10G（SFP+）×8 1G（RJ45）×1（BMC）
最大重量	21.9kg
電源	DC-48V

1.3.3 ソフトウェア要件

本製品は、オープンソースソフトウェアを使用しています。

使用するソフトウェアの一覧とバージョンを以下に示します。

表 1-2 ソフトウェア諸元

OS／ソフトウェア	バージョン	機能
CentOS	8.2	OS
Pacemaker	2.0.4-6	HaaS コントローラ間のクラスタ管理
Corosync	3.0.3	HaaS コントローラ間の死活監視
DRBD	9.13.1	DRBD 専用領域のデータのレプリケーション
Cobbler	3.1.2	IA サーバへの OS インストール
Docker	19.03.13	VM コンテナの基盤
AWX	14.0.0	Web UI/REST API の対応とユーザ管理機能の提供
ansible	2.9.11	ワークフローを自動化
Zabbix	4.0.22	IA サーバの障害監視
GitLab	13.2.4	Ansible Playbook の管理
PostgreSQL	10.14	構成管理用データベース（CMDB）の構築
Nginx	1.19.1	リクエストの振り分け（リバースプロキシ）
Django	3.1	Web アプリケーションフレームワーク

1.4 導入作業の前に

1.4.1 前提条件

導入にあたり、以下の前提条件が満たされている必要があります。

- 物理工事が完了していること。
- 電源が ON 状態であること。
- 『BoxUP 環境定義書』、『BoxUP 環境定義書_IP アドレス』、『BoxUP 環境定義書_ホスト名』を用意すること。
- OS および必要なソフトウェア一式がインストールされていること。

1.4.2 ワークフロー

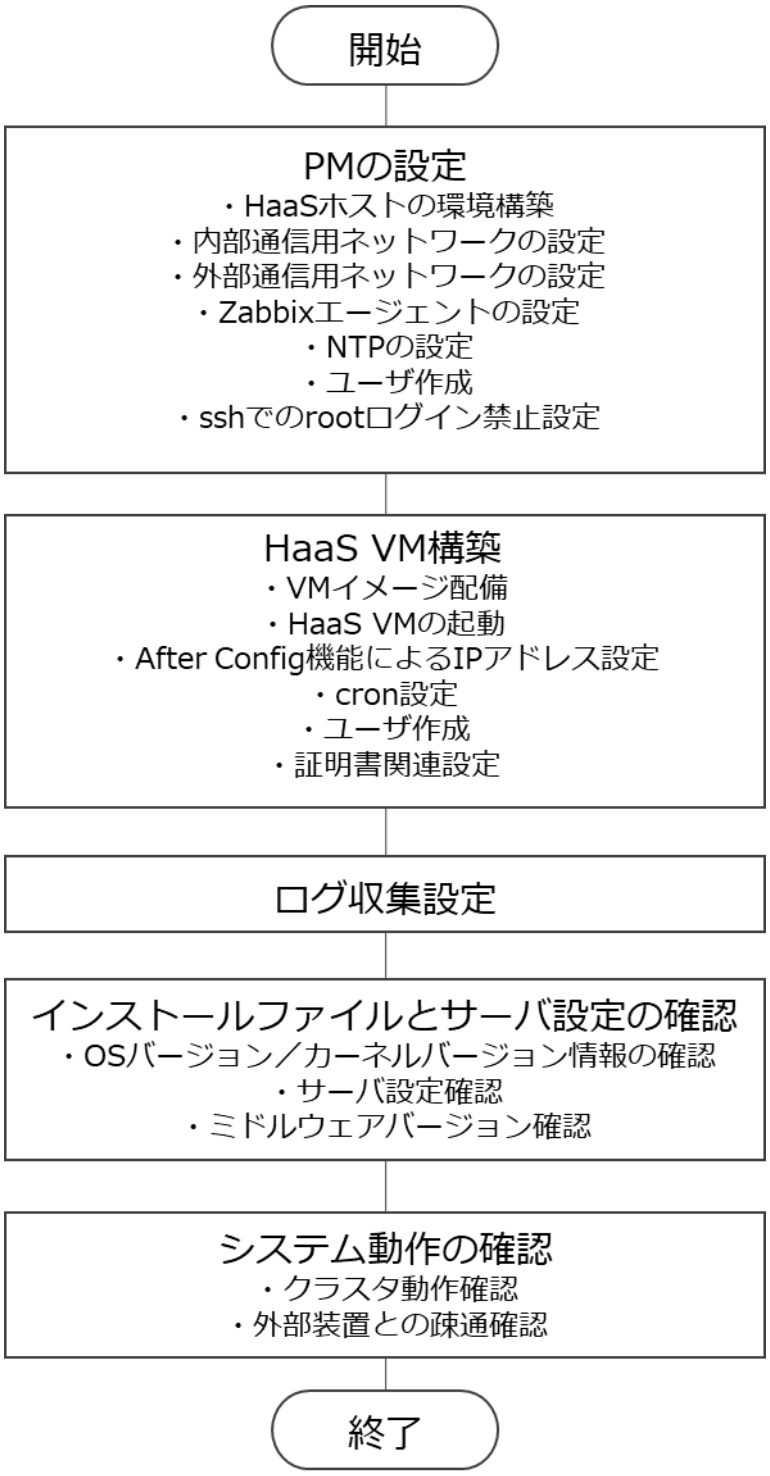

図 1-6 HaaS コントローラ導入のワークフロー

2 PM の設定

本章の手順は全ての HaaS PM に対して実施してください。

■HaaS PM へのログイン

各設定手順では、はじめに HaaS PM へのログインが必要です。

以降に示す手順で HaaS PM にログインする場合はコンソールで接続します。

注： root ユーザを使用し実施してください。

2.1 HaaS ホストの環境構築

HaaS ホストの環境構築手順を以下に示します。

STEP 操作

1 HaaS ホスト名を設定します。

1-1 HaaS ホスト名設定ファイルを更新します。

＜入力フォーマット＞

```
# vi /etc/hostname
```

/etc/hostname ファイルに、設定したいホスト名を記載し保存します。
以下は記載例です。

```
haasctlpm10
```

注1: 既に記載されているホスト名は削除するか、または"#"でコメントアウトします。

注2: ホスト名については『BoxUP 環境定義書_ホスト名』を参照してください。

1-2 設定ファイルを読み込みます。

＜入力フォーマット＞

```
# hostname -F /etc/hostname
```

1-3 設定が反映されていることを確認します。

＜入力フォーマット＞

```
# hostname
```

2.2 内部通信用ネットワークの設定

内部通信用ブリッジの作成手順を以下に示します。

STEP 操作

1 ブリッジ（br-haas-mng）を作成します。

1-1 内部通信用ブリッジ（br-haas-mng）を作成します。

＜入力フォーマット＞

```
# nmcli c add type bridge ifname br-haas-mng con-name br-haas-mng
# nmcli c mod br-haas-mng ipv4.method manual ipv4.addresses "<IP アドレス／プレフィックス>"
# nmcli c up br-haas-mng
```

変数	説明
＜IP アドレス／プレフィックス＞	『BoxUP 環境定義書_IP アドレス』の内部接続を参照してください。

1-2 物理インタフェースをブリッジに接続します。

＜入力フォーマット＞

```
# nmcli c mod eno1 master br-haas-mng
# nmcli c mod eno2 master br-haas-mng
# nmcli c mod eno3 master br-haas-mng
# nmcli c mod eno4 master br-haas-mng
```

1-3 物理インタフェースを up します。

＜入力フォーマット＞

```
# nmcli c up eno1
# nmcli c up eno2
# nmcli c up eno3
# nmcli c up eno4
```

1-4 物理インタフェースの autoconnect 設定を確認します。

＜入力フォーマット＞

```
# nmcli c s eno1 |grep connection.autoconnect:
# nmcli c s eno2 |grep connection.autoconnect:
# nmcli c s eno3 |grep connection.autoconnect:
# nmcli c s eno4 |grep connection.autoconnect:
```

「connection.autoconnect」が「no」となっていることを確認します。

```
connection.autoconnect:                 no
```

1-5 物理インタフェースの autoconnect 設定を yes に設定します。

＜入力フォーマット＞

```
# nmcli c mod eno1 autoconnect yes
# nmcli c mod eno2 autoconnect yes
# nmcli c mod eno3 autoconnect yes
# nmcli c mod eno4 autoconnect yes
```

1-6 物理インタフェースの autoconnect 設定を確認します。

＜入力フォーマット＞

```
# nmcli c s eno1 |grep connection.autoconnect:
# nmcli c s eno2 |grep connection.autoconnect:
# nmcli c s eno3 |grep connection.autoconnect:
# nmcli c s eno4 |grep connection.autoconnect:
```

「connection.autoconnect」が「yes」となっていることを確認します。

```
connection.autoconnect:                 yes
```

2.3 外部通信用ネットワークの設定

外部通信用ブリッジの作成手順を以下に示します。

STEP 操作

1 ブリッジ（br-haas-op）を作成します。

1-1 外部通信用ブリッジ（br-haas-op）を作成します。

＜入力フォーマット＞

```
# nmcli c add type bridge ifname br-haas-op con-name br-haas-op
# nmcli c mod br-haas-op ipv4.method manual ipv4.addresses "<IP アドレス／プレフィックス>"
# nmcli c up br-haas-op
```

変数	説明
＜IP アドレス／プレフィックス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

1-2 VLAN インタフェースを作成します。

＜入力フォーマット＞

```
# nmcli c add type vlan ifname ens1f0.<VLAN ID> dev ens1f0 id <VLAN ID>
# nmcli c add type vlan ifname ens1f1.<VLAN ID> dev ens1f1 id <VLAN ID>
# nmcli c add type vlan ifname ens2f0.<VLAN ID> dev ens2f0 id <VLAN ID>
# nmcli c add type vlan ifname ens2f1.<VLAN ID> dev ens2f1 id <VLAN ID>
```

変数	説明
＜VLAN ID＞	外部接続の VLAN ID 『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

1-3 作成したインタフェースをブリッジに接続します。

＜入力フォーマット＞

```
# nmcli c mod vlan-ens1f0.<VLAN ID> master br-haas-op
# nmcli c mod vlan-ens1f1.<VLAN ID> master br-haas-op
# nmcli c mod vlan-ens2f0.<VLAN ID> master br-haas-op
# nmcli c mod vlan-ens2f1.<VLAN ID> master br-haas-op
```

変数	説明
＜VLAN ID＞	外部接続の VLAN ID 『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

1-4 作成したインタフェースを up します。

＜入力フォーマット＞

```
# nmcli c up vlan-ens1f0.＜VLAN ID＞
# nmcli c up vlan-ens1f1.＜VLAN ID＞
# nmcli c up vlan-ens2f0.＜VLAN ID＞
# nmcli c up vlan-ens2f1.＜VLAN ID＞
```

変数	説明
＜VLAN ID＞	外部接続の VLAN ID 『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

2 外部通信用ネットワークの default gateway 設定を行います。

2-1 default gateway を設定します。

＜入力フォーマット＞

```
# nmcli c m br-haas-op ipv4.gateway "＜nexthop の IP アドレス＞"
```

変数	説明
＜nexthop の IP アドレス＞	O&M 専用ルータの nexthop の IP アドレス

2-2 設定を反映します。

＜入力フォーマット＞

```
# nmcli c reload
# nmcli c down br-haas-op
# nmcli c up br-haas-op
```

2.4 Zabbix Agent の設定

Zabbix Agent の設定手順を以下に示します。

STEP	操作

1 Zabbix Agent のバージョン情報を確認します。

＜入力フォーマット＞

```
# zabbix_agentd --version
```

「4.0.22」であることを確認します。

```
zabbix_agentd (daemon) (Zabbix) 4.0.22
Revision 073cb9f 29 June 2020, compilation time: Jun 29 2020 14:13:44

Copyright (C) 2020 Zabbix SIA
License GPLv2+: GNU GPL version 2 or later
<http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it
according to
the license. There is NO WARRANTY, to the extent permitted by law.

This product includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit (http://www.openssl.org/).

Compiled with OpenSSL 1.0.2k-fips  26 Jan 2017
Running with OpenSSL 1.0.2k-fips  26 Jan 2017
```

2 SELinux を無効化します。

2-1 SELinux の設定状況を確認します。

＜入力フォーマット＞

```
# getenforce
```

「Disabled」と表示される場合

SELinux は既に無効化されているため、STEP 3 に進みます。

```
Disabled
```

「Enforcing」と表示される場合

SELinux が有効化されています。無効化するために STEP 2-2 に進みます。

```
Enforcing
```

2-2 設定ファイルを編集し、SELinux を無効化します。

＜入力フォーマット＞

```
# vi /etc/selinux/config
```

以下となるように設定します（★印部分）。

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled  ★
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected
processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

2-3 再起動します。

＜入力フォーマット＞

```
# shutdown -r now
```

2-4 再度、SELinux の設定状況を確認します。

＜入力フォーマット＞

```
# getenforce
```

「Disabled」と表示されることを確認します。

```
Disabled
```

3 firewall 設定を行います。

＜入力フォーマット＞

```
# firewall-cmd --add-port=10050/tcp --permanent
# firewall-cmd --reload
# firewall-cmd --list-all
```

4 設定ファイルを修正します。

4-1 Zabbix Agent の設定ファイルを修正します。

＜入力フォーマット＞

```
# vi /etc/zabbix/zabbix_agentd.conf
```

以下となるように設定します（★印部分）。

```
# This is a configuration file for Zabbix agent daemon (Unix)
# To get more information about Zabbix, visit http://www.zabbix.com

############ GENERAL PARAMETERS #################

### Option: PidFile
#       Name of PID file.
#
# Mandatory: no
# Default:
# PidFile=/tmp/zabbix_agentd.pid

PidFile=/var/run/zabbix/zabbix_agentd.pid ★

### Option: LogType
#       Specifies where log messages are written to:
#               system  - syslog
#               file    - file specified with LogFile parameter
#               console - standard output
#
# Mandatory: no
# Default:
# LogType=file

### Option: LogFile
#       Log file name for LogType 'file' parameter.
#
# Mandatory: yes, if LogType is set to file, otherwise no
# Default:
# LogFile=

LogFile=/var/log/zabbix/zabbix_agentd.log ★

### Option: LogFileSize
#       Maximum size of log file in MB.
#       0 - disable automatic log rotation.
#
# Mandatory: no
# Range: 0-1024
# Default:
# LogFileSize=1
```

```
LogFileSize=0 ★

### Option: DebugLevel
#       Specifies debug level:
#       0 - basic information about starting and stopping of Zabbix
processes
#       1 - critical information
#       2 - error information
#       3 - warnings
#       4 - for debugging (produces lots of information)
#       5 - extended debugging (produces even more information)
#
# Mandatory: no
# Range: 0-5
# Default:
# DebugLevel=3

### Option: SourceIP
#       Source IP address for outgoing connections.
#
# Mandatory: no
# Default:
# SourceIP=

### Option: EnableRemoteCommands
#       Whether remote commands from Zabbix server are allowed.
#       0 - not allowed
#       1 - allowed
#
# Mandatory: no
# Default:
# EnableRemoteCommands=0

EnableRemoteCommands=1 ★

### Option: LogRemoteCommands
#       Enable logging of executed shell commands as warnings.
#       0 - disabled
#       1 - enabled
#
# Mandatory: no
# Default:
# LogRemoteCommands=0

##### Passive checks related

### Option: Server
```

```
#       List of comma delimited IP addresses, optionally in CIDR notation,
or DNS names of Zabbix servers and Zabbix proxies.
#       Incoming connections will be accepted only from the hosts listed
here.
#       If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1',
'::ffff:127.0.0.1' are treated equally
#       and '::/0' will allow any IPv4 or IPv6 address.
#       '0.0.0.0/0' can be used to allow any IPv4 address.
#       Example:
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=0.0.0.0/0 ★

### Option: ListenPort
#       Agent will listen on this port for connections from the server.
#
# Mandatory: no
# Range: 1024-32767
# Default:
# ListenPort=10050

### Option: ListenIP
#       List of comma delimited IP addresses that the agent should listen
on.
#       First IP address is sent to Zabbix server if connecting to it to
retrieve list of active checks.
#
# Mandatory: no
# Default:
# ListenIP=0.0.0.0

### Option: StartAgents
#       Number of pre-forked instances of zabbix_agentd that process passive
checks.
#       If set to 0, disables passive checks and the agent will not listen
on any TCP port.
#
# Mandatory: no
# Range: 0-100
# Default:
# StartAgents=3

##### Active checks related

### Option: ServerActive
```

```
#       List of comma delimited IP:port (or DNS name:port) pairs of Zabbix
servers and Zabbix proxies for active checks.
#       If port is not specified, default port is used.
#       IPv6 addresses must be enclosed in square brackets if port for that
host is specified.
#       If port is not specified, square brackets for IPv6 addresses are
optional.
#       If this parameter is not specified, active checks are disabled.
#       Example:
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=<内部通信用 NW の VIP> ★

### Option: Hostname
#       Unique, case sensitive hostname.
#       Required for active checks and must match hostname as configured on
the server.
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

#Hostname=Zabbix server ★"#"でコメントアウト

### Option: HostnameItem
#       Item used for generating Hostname if it is undefined. Ignored if
Hostname is defined.
#       Does not support UserParameters or aliases.
#
# Mandatory: no
# Default:
# HostnameItem=system.hostname

HostnameItem=system.hostname ★

### Option: HostMetadata
#       Optional parameter that defines host metadata.
#       Host metadata is used at host auto-registration process.
#       An agent will issue an error and not start if the value is over
limit of 255 characters.
#       If not defined, value will be acquired from HostMetadataItem.
#
# Mandatory: no
# Range: 0-255 characters
```

```
# Default:
# HostMetadata=

HostMetadata=HaaSPM ★

### Option: HostMetadataItem
#       Optional parameter that defines an item used for getting host
metadata.
#       Host metadata is used at host auto-registration process.
#       During an auto-registration request an agent will log a warning
message if
#       the value returned by specified item is over limit of 255
characters.
#       This option is only used when HostMetadata is not defined.
#
# Mandatory: no
# Default:
# HostMetadataItem=

### Option: RefreshActiveChecks
#       How often list of active checks is refreshed, in seconds.
#
# Mandatory: no
# Range: 60-3600
# Default:
# RefreshActiveChecks=120

RefreshActiveChecks=120 ★

### Option: BufferSend
#       Do not keep data longer than N seconds in buffer.
#
# Mandatory: no
# Range: 1-3600
# Default:
# BufferSend=5

### Option: BufferSize
#       Maximum number of values in a memory buffer. The agent will send
#       all collected data to Zabbix Server or Proxy if the buffer is full.
#
# Mandatory: no
# Range: 2-65535
# Default:
# BufferSize=100

### Option: MaxLinesPerSecond
#       Maximum number of new lines the agent will send per second to Zabbix
Server
```

```
#       or Proxy processing 'log' and 'logrt' active checks.
#       The provided value will be overridden by the parameter 'maxlines',
#       provided in 'log' or 'logrt' item keys.
#
# Mandatory: no
# Range: 1-1000
# Default:
# MaxLinesPerSecond=20

############ ADVANCED PARAMETERS #################

### Option: Alias
#       Sets an alias for an item key. It can be used to substitute long and
complex item key with a smaller and simpler one.
#       Multiple Alias parameters may be present. Multiple parameters with
the same Alias key are not allowed.
#       Different Alias keys may reference the same item key.
#       For example, to retrieve the ID of user 'zabbix':
#       Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,^zabbix:.:([0-
9]+),,,,¥1]
#       Now shorthand key zabbix.userid may be used to retrieve data.
#       Aliases can be used in HostMetadataItem but not in HostnameItem
parameters.
#
# Mandatory: no
# Range:
# Default:

### Option: Timeout
#       Spend no more than Timeout seconds on processing
#
# Mandatory: no
# Range: 1-30
# Default:
# Timeout=3

Timeout=20 ★

### Option: AllowRoot
#       Allow the agent to run as 'root'. If disabled and the agent is
started by 'root', the agent
#       will try to switch to the user specified by the User configuration
option instead.
#       Has no effect if started under a regular user.
#       0 - do not allow
#       1 - allow
#
# Mandatory: no
# Default:
```

```
# AllowRoot=0

AllowRoot=1 ★

### Option: User
#       Drop privileges to a specific, existing user on the system.
#       Only has effect if run as 'root' and AllowRoot is disabled.
#
# Mandatory: no
# Default:
# User=zabbix

### Option: Include
#       You may include individual files or all files in a directory in the
configuration file.
#       Installing Zabbix will create include directory in /usr/local/etc,
unless modified during the compile time.
#
# Mandatory: no
# Default:
# Include=

Include=/etc/zabbix/zabbix_agentd.d/*.conf ★

# Include=/usr/local/etc/zabbix_agentd.userparams.conf
# Include=/usr/local/etc/zabbix_agentd.conf.d/
# Include=/usr/local/etc/zabbix_agentd.conf.d/*.conf

####### USER-DEFINED MONITORED PARAMETERS #######

### Option: UnsafeUserParameters
#       Allow all characters to be passed in arguments to user-defined
parameters.
#       The following characters are not allowed:
#       ¥ ' " ` * ? [ ] { } ~ $ ! & ; ( ) < > | # @
#       Additionally, newline characters are not allowed.
#       0 - do not allow
#       1 - allow
#
# Mandatory: no
# Range: 0-1
# Default:
# UnsafeUserParameters=0

UnsafeUserParameters=1 ★

### Option: UserParameter
#       User-defined parameter to monitor. There can be several user-defined
parameters.
```

```
#       Format: UserParameter=<key>,<shell command>
#       See 'zabbix_agentd' directory for examples.
#
# Mandatory: no
# Default:
# UserParameter=

####### LOADABLE MODULES #######

### Option: LoadModulePath
#       Full path to location of agent modules.
#       Default depends on compilation options.
#       To see the default path run command "zabbix_agentd --help".
#
# Mandatory: no
# Default:
# LoadModulePath=${libdir}/modules

### Option: LoadModule
#       Module to load at agent startup. Modules are used to extend
functionality of the agent.
#       Formats:
#               LoadModule=<module.so>
#               LoadModule=<path/module.so>
#               LoadModule=</abs_path/module.so>
#       Either the module must be located in directory specified by
LoadModulePath or the path must precede the module name.
#       If the preceding path is absolute (starts with '/') then
LoadModulePath is ignored.
#       It is allowed to include multiple LoadModule parameters.
#
# Mandatory: no
# Default:
# LoadModule=

####### TLS-RELATED PARAMETERS #######

### Option: TLSConnect
#       How the agent should connect to server or proxy. Used for active
checks.
#       Only one value can be specified:
#               unencrypted - connect without encryption
#               psk         - connect using TLS and a pre-shared key
#               cert        - connect using TLS and a certificate
#
# Mandatory: yes, if TLS certificate or PSK parameters are defined (even
for 'unencrypted' connection)
# Default:
# TLSConnect=unencrypted
```

```
### Option: TLSAccept
#       What incoming connections to accept.
#       Multiple values can be specified, separated by comma:
#             unencrypted - accept connections without encryption
#             psk         - accept connections secured with TLS and a pre-
shared key
#             cert        - accept connections secured with TLS and a
certificate
#
# Mandatory: yes, if TLS certificate or PSK parameters are defined (even
for 'unencrypted' connection)
# Default:
# TLSAccept=unencrypted

### Option: TLSCAFile
#       Full pathname of a file containing the top-level CA(s) certificates
for
#       peer certificate verification.
#
# Mandatory: no
# Default:
# TLSCAFile=

### Option: TLSCRLFile
#       Full pathname of a file containing revoked certificates.
#
# Mandatory: no
# Default:
# TLSCRLFile=

### Option: TLSServerCertIssuer
#      Allowed server certificate issuer.
#
# Mandatory: no
# Default:
# TLSServerCertIssuer=

### Option: TLSServerCertSubject
#      Allowed server certificate subject.
#
# Mandatory: no
# Default:
# TLSServerCertSubject=

### Option: TLSCertFile
#       Full pathname of a file containing the agent certificate or
certificate chain.
#
```

```
# Mandatory: no
# Default:
# TLSCertFile=

### Option: TLSKeyFile
#       Full pathname of a file containing the agent private key.
#
# Mandatory: no
# Default:
# TLSKeyFile=

### Option: TLSPSKIdentity
#       Unique, case sensitive string used to identify the pre-shared key.
#
# Mandatory: no
# Default:
# TLSPSKIdentity=

### Option: TLSPSKFile
#       Full pathname of a file containing the pre-shared key.
#
# Mandatory: no
# Default:
# TLSPSKFile=

####### For advanced users - TLS ciphersuite selection criteria #######

### Option: TLSCipherCert13
#       Cipher string for OpenSSL 1.1.1 or newer in TLS 1.3.
#       Override the default ciphersuite selection criteria for certificate-
based encryption.
#
# Mandatory: no
# Default:
# TLSCipherCert13=

### Option: TLSCipherCert
#       GnuTLS priority string or OpenSSL (TLS 1.2) cipher string.
#       Override the default ciphersuite selection criteria for certificate-
based encryption.
#       Example for GnuTLS:
#               NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-
CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
#       Example for OpenSSL:
#               EECDH+aRSA+AES128:RSA+aRSA+AES128
#
# Mandatory: no
# Default:
# TLSCipherCert=
```

```
### Option: TLSCipherPSK13
#       Cipher string for OpenSSL 1.1.1 or newer in TLS 1.3.
#       Override the default ciphersuite selection criteria for PSK-based
encryption.
#       Example:
#               TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
#
# Mandatory: no
# Default:
# TLSCipherPSK13=

### Option: TLSCipherPSK
#       GnuTLS priority string or OpenSSL (TLS 1.2) cipher string.
#       Override the default ciphersuite selection criteria for PSK-based
encryption.
#       Example for GnuTLS:
#               NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-
CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
#       Example for OpenSSL:
#               kECDHEPSK+AES128:kPSK+AES128
#
# Mandatory: no
# Default:
# TLSCipherPSK=

### Option: TLSCipherAll13
#       Cipher string for OpenSSL 1.1.1 or newer in TLS 1.3.
#       Override the default ciphersuite selection criteria for certificate-
and PSK-based encryption.
#       Example:
#
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
#
# Mandatory: no
# Default:
# TLSCipherAll13=

### Option: TLSCipherAll
#       GnuTLS priority string or OpenSSL (TLS 1.2) cipher string.
#       Override the default ciphersuite selection criteria for certificate-
and PSK-based encryption.
#       Example for GnuTLS:
#               NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-
GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-
ALL:+CTYPE-X.509
#       Example for OpenSSL:
#
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

```
#
# Mandatory: no
# Default:
# TLSCipherAll=

#UserParameter ★
# common ★
UserParameter=hostname_check,hostname >/dev/null 2>&1;echo $? ★
UserParameter=ping[*],ping -c1 -w 25 $1 >/dev/null 2>&1;echo $? ★

# crond ★
UserParameter=crond.ps_sys,ps -ef | grep crond | awk '{print $3}' | grep
"^1$" | wc -l ★

# PID Check ★
UserParameter=prc_chk.pid[*],pgrep $1 | head -1 ★
UserParameter=container.chk[*],docker inspect $1 2>/dev/null | grep -e
'"Status": "running"' | wc -l ★
UserParameter=psc_status.chk[*],pcs status 2>/dev/null | grep $1 | grep
"Started"| wc -l ★
UserParameter=psc_status.pcs,pcs status 2>/dev/null | grep Online | wc -l
★
```

変数	説明
＜内部通信用 NW の VIP＞	『BoxUP 環境定義書_IP アドレス』の内部接続を参照してください。

4-2 Zabbix Agent の service ファイルを修正します。

```
#vi /usr/lib/systemd/system/zabbix-agent.service
```

以下となるように修正します（★印部分）。

```
[Service]
#User=zabbix    ★"#"でコメントアウト
#Group=zabbix   ★"#"でコメントアウト
```

5 Zabbix Agent を再起動します。

＜入力フォーマット＞

```
# systemctl daemon-reload
# systemctl restart zabbix-agent
# systemctl status zabbix-agent
```

エラーがなく、「Active: active (running)」と表示されることを確認します。

```
● zabbix-agent.service - Zabbix Agent
   Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; disabled; vendor
preset: disabled)
   Active: active (running) since Wed 2020-11-11 21:52:41 JST; 2s ago
  Process: 2271 ExecStart=/usr/sbin/zabbix_agentd -c $CONFFILE (code=exited,
status=0/SUCCESS)
 Main PID: 2273 (zabbix_agentd)
    Tasks: 6 (limit: 101391)
   Memory: 4.1M
   CGroup: /system.slice/zabbix-agent.service
           tq2273 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
           tq2274 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
           tq2275 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
           tq2276 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
           tq2277 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
           mq2278 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

11月 11 21:52:41 haas2 systemd[1]: Starting Zabbix Agent...
11月 11 21:52:41 haas2 systemd[1]: zabbix-agent.service: Supervising process 2273
which is not our child. We'll most likely not notice when it exits.
11月 11 21:52:41 haas2 systemd[1]: Started Zabbix Agent.
```

6 Zabbix Agent の自動起動設定をします。

＜入力フォーマット＞

```
# systemctl enable zabbix-agent
# systemctl is-enabled zabbix-agent
```

「enabled」と表示されることを確認します。

```
# systemctl is-enabled zabbix-agent
enabled
```

2.5 NTP の設定

NTP の設定手順を以下に示します。

STEP 操作

1 設定ファイルを修正します。

＜入力フォーマット＞

```
# vi /etc/chrony.conf
```

以下となるように設定します（★印部分）。

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#pool 2.centos.pool.ntp.org iburst  ★"#"でコメントアウト
server <NTP サーバ 1> iburst  ★
server <NTP サーバ 2> iburst  ★
  :（省略）
```

変数	説明
＜NTP サーバ 1＞、 ＜NTP サーバ 2＞	NTP サーバ 1 および NTP サーバ 2 の IP アドレス

2 chronyd の起動と自動起動設定をします。

2-1 chronyd を起動します。

＜入力フォーマット＞

```
# systemctl start chronyd
# systemctl status chronyd
```

エラーがなく、「active (running)」と表示されることを確認します。

```
● chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled;
vendor preset: enabled)
   Active: active (running) since Wed 2021-05-19 11:07:33 JST; 1min
59s ago
     Docs: man:chronyd(8)
           man:chrony.conf(5)
  Process: 904 ExecStartPost=/usr/libexec/chrony-helper update-
daemon (code=exited, status=0/SUCCESS)
  Process: 886 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited,
status=0/SUCCESS)
 Main PID: 899 (chronyd)
    Tasks: 1 (limit: 117261)
```

```
   Memory: 2.0M
   CGroup: /system.slice/chronyd.service
           mq899 /usr/sbin/chronyd

 5月 19 11:07:33 haas1 systemd[1]: Starting NTP client/server...
 5月 19 11:07:33 haas1 chronyd[899]: chronyd version 3.5 starting
(+CMDMON +NTP +REFCLOCK +RTC +PRIVDROP +SCFILTER +SIGND +ASYNCDNS
+SECHASH +IPV6 +DEBUG)
 5月 19 11:07:33 haas1 chronyd[899]: Frequency 0.000 +/-
1000000.000 ppm read from /var/lib/chrony/drift
 5月 19 11:07:33 haas1 chronyd[899]: Using right/UTC timezone to
obtain leap second data
 5月 19 11:07:33 haas1 systemd[1]: Started NTP client/server.
```

2-2 chronyd の自動起動設定をします。

＜入力フォーマット＞

```
# systemctl enable chronyd
# systemctl is-enabled chronyd
```

「enabled」と表示されることを確認します。

```
# systemctl is-enabled chronyd
enabled
```

2.6 ユーザ作成

次項で ssh での root ログイン禁止設定を実施するため、一般ユーザの作成を行います。

参考： ユーザ作成の詳細については、『HaaS コントローラ操作マニュアル』の「2.2.1.1　ユーザの管理操作」の「■ユーザの作成」を参照してください。

2.7 ssh での root ログイン禁止設定

ssh での root ログインを禁止するための設定手順を以下に示します。

STEP	操作

1 sshd_config ファイルを修正します。

＜入力フォーマット＞

```
# cd /etc/ssh
# vi sshd_config
```

以下となるように修正します（★印部分）。

```
          :（省略）
#PermitRootLogin yes   ★"#"でコメントアウト
PermitRootLogin no   ★
          :（省略）
```

2 sshd を再起動します。

＜入力フォーマット＞

```
# systemctl restart sshd.service
# systemctl status sshd.service
```

エラーがなく、「Active: active (running)」と表示されることを確認します。

```
# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor
preset: enabled)
   Active: active (running) since Tue 2021-11-16 22:56:34 JST; 1min 37s
ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 40244 (sshd)
    Tasks: 1 (limit: 410379)
   Memory: 1.2M
   CGroup: /system.slice/sshd.service
           mq40244 /usr/sbin/sshd -D -oCiphers=aes256-
gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-
cbc,aes128-gcm@openssh.com,aes128-ctr>
:
```

3 HaaS VM 構築

3.1 構築要件

HaaS VM の構築要件は以下のとおりです。

- 2 章の設定を実施した物理サーバがあること。
- HaaS VM イメージファイルと XML ファイルのチェックサム値を取得していること。
- Zabbix の設定に伴い、以下アラームが発生します。

アラーム ID	アラーム名	障害の深刻度及び障害の説明
D14A261	Service terminated Zabbix_API	重度障害、ZabbixAPI のサービス断 初回起動時のみ発報、2 分で復旧します
D14B011	Service terminated AWX_API	重度障害、AWXAPI のサービス断 初回起動時のみ発報、2 分で復旧します
D14B021	Service terminated CMDB_API	重度障害、CMDBAPI のサービス断 初回起動時のみ発報、2 分で復旧します
D14B031	Service terminated Gitlab_API	軽度障害、GitlabAPI のサービス断 初回起動時のみ発報、2 分で復旧します

3.2 HaaS VM イメージ配備

HaaS VM イメージファイルと XML ファイルの転送と格納を実施します。

本手順は全ての HaaS PM 上で実施します。

STEP	操作
1	HaaS PM にログインします。 HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。
2	HaaS VM イメージを格納するディレクトリを作成します。 ＜入力フォーマット＞ `# mkdir /home/images`
3	HaaS VM イメージファイルと XML ファイルを HaaS PM に転送します。 HaaS VM イメージファイルと XML ファイルをそれぞれの格納先ディレクトリに転送します。 HaaS VM イメージファイル：/home/images XML ファイル：/tmp

4 イメージファイルが格納されたことを確認します。

4-1 格納先のイメージファイルを確認します。

＜入力フォーマット＞

```
# cd /home/images
# ls ＜イメージファイル名＞
```

変数	説明
＜イメージファイル名＞	格納先のイメージファイル名 例：haasctl_xxxx_comp.qcow2

格納先にイメージファイルが格納されていることを確認します。

```
# ls haasctl_xxxx_comp.qcow2
haasctl_xxxx_comp.qcow2
```

4-2 格納先 XML ファイルを確認します。

＜入力フォーマット＞

```
# cd /tmp
# ls ＜XML ファイル名＞
```

変数	説明
＜XML ファイル名＞	Haas VM の XML ファイル名 例：haasctl_xxxx.xml

格納先に XML ファイルが格納されていることを確認します。

```
# ls haasctl_xxxx.xml
haasctl_xxxx.xml
```

4-3 チェックサム値を確認します。

＜入力フォーマット＞

```
# cd /home/images
# md5sum ＜イメージファイル名＞
# cd /tmp
# md5sum ＜XML ファイル名＞
```

変数	説明
＜イメージファイル名＞	イメージファイル名 例：haasctl_xxxx_comp.qcow2
＜XML ファイル名＞	Haas VM の XML ファイル名 例：haasctl_xxxx.xml

イメージファイルおよび XML ファイルのチェックサム値が、転送前の各ファイルのチェックサム値と同じであることを確認します。

```
# md5sum haasctl_xxxx_comp.qcow2
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx haasctl_xxxx_comp.qcow2

# md5sum haasctl_xxxx.xml
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx haasctl_xxxx.xml
```

変数	説明
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	指定されたファイルのチェックサム値（16 進数表記、32 桁）を表示します。

5 VM イメージファイルを非圧縮状態に戻します。

＜入力フォーマット＞

```
# cd /home/images
# qemu-img convert -f qcow2 -O qcow2 ＜圧縮ファイル名＞ ＜非圧縮ファイル名＞
```

変数	説明
＜圧縮ファイル名＞	圧縮状態のイメージファイル名 例：haasctl_xxxx_comp.qcow2
＜非圧縮ファイル名＞	圧縮解除後のイメージファイル名 例：0 系の場合　haasctl00_xxxx.qcow2 1 系の場合　haasctl01_xxxx.qcow2 2 系の場合　haasctl02_xxxx.qcow2

注： イメージファイルは共通のため、圧縮解除後のイメージファイル名は、0 系、1 系、2 系が分かるようにしてください。

6 XML ファイル名を変更します。

XML ファイルは共通のため、0 系、1 系、2 系が分かるように XML ファイル名を変更します。

＜入力フォーマット＞

```
# cd /tmp
# mv ＜変更前の XML ファイル名＞ ＜変更後の XML ファイル名＞
```

変数	説明
＜変更前の XML ファイル名＞	変更前の XML ファイル名 例：haasctl_xxxx.xml
＜変更後の XML ファイル名＞	変更後の XML ファイル名 例：0 系の場合　haasctl00_xxxx.xml 1 系の場合　haasctl01_xxxx.xml 2 系の場合　haasctl02_xxxx.xml

7 XML ファイルを修正します。

＜入力フォーマット＞

```
# vi / tmp/＜XML ファイル名＞
```

変数	説明
＜XML ファイル名＞	Haas VM の XML ファイル名 例：haasctl00_xxxx.xml

以下の★印部分の VM 名を修正します。

```
<domain type='kvm'>
  <name>＜VM 名＞</name>  ★
  <memory unit='KiB'>67108864</memory>
  <currentMemory unit='KiB'>67108864</currentMemory>
  <vcpu placement='static'>44</vcpu>
```

変数	説明
＜VM 名＞	Haas VM のホスト名 『BoxUP 環境定義書_ホスト名』の HaaS VM の Hostname を参照してください。

以下の★印部分の VM イメージのファイルパスを修正します。

```
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/home/images/＜非圧縮ファイル名＞'/>  ★
      <target dev='vda' bus='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </disk>
```

変数	説明
＜非圧縮ファイル名＞	STEP 5 の圧縮解除後のイメージファイル名 例：0 系の場合　haasctl00_xxxx.qcow2 1 系の場合　haasctl01_xxxx.qcow2 2 系の場合　haasctl02_xxxx.qcow2

以下の★印部分の MAC アドレスを修正します。

注：　各 HaaS-VM で MAC アドレスが重複しないように設定します。
MAC アドレスが重複すると HaaS VM 間や対向ノードと通信が出来なくなる可能性があります。

```
<interface type='bridge'>
  <mac address='＜MAC アドレス＞'/>  ★
  <source bridge='br-haas-mng'/>
  <model type='e1000'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
</interface>
<interface type='bridge'>
  <mac address='＜MAC アドレス＞'/>  ★
  <source bridge='br-haas-op'/>
  <model type='e1000'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

MAC アドレスの修正は以下の例を参考にしてください。

例）リリース時の XML で以下のように記載されている場合
<mac address='52:54:00:51:00:02'/>

0 系の場合：<mac address='52:54:00:51:**00**:02'/>
1 系の場合：<mac address='52:54:00:51:**01**:02'/>
2 系の場合：<mac address='52:54:00:51:**02**:02'/>

3.3 HaaS VM の起動

「3.2 HaaS VM イメージ配備」に続いて、HaaS VM の登録・起動を行います。

STEP 操作

1 HaaS PM にログインします。

HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。

2 VM を登録します。

＜入力フォーマット＞

```
# virsh define /tmp/＜XML ファイル名＞
```

変数	説明
＜XML ファイル名＞	Haas VM の XML ファイル名 例： haasctl00_xxxx.xml

```
# virsh define /tmp/haasctl00_xxxx.xml

ドメイン ccc...c が /tmp/haasctl00_xxxx.xml から定義されました
```

変数	説明
ccc...c	指定した XML ファイルで定義された VM 名

3 VM を起動します。

＜入力フォーマット＞

```
# virsh start ＜VM 名＞
```

変数	説明
＜VM 名＞	STEP 2 で登録した Haas VM の VM 名

```
# virsh start ccc...c

ドメイン ccc...c が起動されました
```

変数	説明
ccc...c	Haas VM の VM 名

3.4 After config 機能

「3.3 HaaS VM の起動」に続いて、HaaS VM の IP アドレス設定を行います。

After config 機能により HaaS VM の起動後に、環境に合わせた IP アドレスを設定することができます。

STEP	操作

1 インタフェースの自動起動を ON にします。

以下の手順（1-1～1-3）を haas0～haas2 に対して実施します。

1-1 HaaS PM から HaaS VM にコンソールログインします。

HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。

＜入力フォーマット＞

```
# virsh console <VM 名>
```

注: After config 実施前はどの VM にログインしてもコンソール上は「haas0」と表示されます。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

注: “Connected to domain ＜VM 名＞ Escape character is ^]”が表示されたまま画面が遷移しない場合、”Enter”キーを押下し、ログインしてください。

1-2 インタフェースの自動起動を ON にします。

＜入力フォーマット＞

```
# nmcli c mod ens2 autoconnect yes
# nmcli c mod ens3 autoconnect yes
```

1-3 コンソールからログアウトします。

「Ctrl」キーと「]」キーを同時に押します。

2 After config（nmcli_change）を実行します。

以下の手順（2-1～2-8）を haas0～haas2 に対して実施します。

2-1 HaaS PM から HaaS VM にコンソールログインします。

HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。

＜入力フォーマット＞

```
# virsh console <VM 名>
```

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

注: “Connected to domain ＜VM 名＞ Escape character is ^]”が表示されたまま画面が遷移しない場合、”Enter”キーを押下し、ログインしてください。

2-2 インベントリファイルを修正します。

＜入力フォーマット＞

```
# cd /root/after_config/inventories/
# vi 01_interface_ip_address_change_inventory.yml
```

以下を参考に修正します。

```
# vi 01_interface_ip_address_change_inventory.yml
all:
  vars:
    hostname: ＜ホスト名＞

    ha_eth_count: ＜IF数＞

    network:
    - # management_nw:
      conn_name: ens2 ★設定変更不要
      ip_address: ＜内部NWのIP＞
      prefix: ＜内部NWのprefix＞
      dns: ＜DNSサーバのIP＞
    - # operation_nw:
      conn_name: ens3 ★設定変更不要
      ip_address: ＜外部NWのIP＞
      prefix: ＜外部NWのprefix＞
      dns: ＜DNSサーバのIP＞
#    - # extoperation_nw:
#      conn_name:       ens7              # Conditional
#      ip_address:      192.168.60.16     # Conditional
#      prefix:          24                # Conditional
#      dns:             8.8.8.8           # Optional

    default_gateway:
      conn_name: ens3 ★設定変更不要
      gateway: ＜外部NWのgateway＞

    static_route:
      conn_name:       ens7              # Optional
      network_address:  192.168.60.0     # Optional
      prefix:          24                # Optional
      nexthop:         192.168.60.1      # Optional

    initialize_network:
    - conn_name: ens2 ★設定変更不要
    - conn_name: ens3 ★設定変更不要
```

変数	説明
＜ホスト名＞	対象のホスト名（設定必須）
＜IF 数＞	「2」を指定（ネットワーク数）

変数	説明
＜内部 NW の IP＞	内部通信用 NW の IP アドレス
＜内部 NW の prefix＞	内部通信用 NW の prefix
＜DNS サーバの IP＞	DNS サーバの IP アドレス 複数ある場合は、カンマ（,）区切りで設定できます。
＜外部 NW の IP＞	外部通信用 NW の IP アドレス
＜外部 NW の prefix＞	外部通信用 NW の prefix
＜外部 NW の gateway＞	外部通信用 NW の gateway IP アドレス

注: ホスト名や IP アドレスなどの値の詳細については、『BoxUP 環境定義書_ホスト名』および『BoxUP 環境定義書_IP アドレス』を参照してください。

2-3 IP アドレス書き換え前の確認を行います。

＜入力フォーマット＞

```
# ip a
```

2-4 After config を実行します。

＜入力フォーマット＞

```
# cd /root/after_config
# ./01_interface_ip_address_change.sh
```

01_interface_ip_address_change.sh は再起動処理が含まれております。再起動完了後に「login」と表示される事を確認します。以下は表示例です。

```
# ./01_interface_ip_address_change.sh
    :（省略）
haasctl00 login:
```

注: 「To Abort waiting enter 'Yes'」と表示された場合は、「Yes」を入力し、Enter を押下してください。

login 表示確認後、そのままログインして下さい。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2-5 IP アドレスが変更されていることを確認します。

＜入力フォーマット＞

```
# ip a
```

2-6 コンソールからログアウトします。

「Ctrl」キーと「]」キーを同時に押します。

2-7 HaaS VM にログインします。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 IP アドレス＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

ログインできることを確認します。

2-8 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注: ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中の PM、VM から抜けるまで繰り返し実施してください。

3 After config（package_change）を実行します。

以下の手順（3-1～3-6）は haas0 に対してのみ実施します。

3-1 HaaS VM にログインします。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 IP アドレス＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

3-2 **インベントリファイルを修正します。**

＜入力フォーマット＞

```
# cd /root/after_config/inventories/
# vi 02_construction_change_inventory.yml
```

以下のように修正します。

```
# vi 02_construction_change_inventory.yml
all:
  hosts:
    haas.0:  ★設定変更不要
    haas.1:  ★設定変更不要
    haas.2:  ★設定変更不要

  vars:

    ha_node_count:        <HaaS VM 数>

    ha_eth_count:         <IF 数>

    drbd_var:
      drbd_disk:          /dev/vda2  ★Default 値のままで設定不要

    hostname:
      node0:              <haas0 系のホスト名>
      node1:              <haas1 系のホスト名>
      node2:              <haas2 系のホスト名>

    domain_ip_relations:
      mng_vip:            <内部 NW の VIP>
      haas0:              <haas0 系の内部 NW の IP>
      haas1:              <haas1 系の内部 NW の IP>
      haas2:              <haas2 系の内部 NW の IP>
      op_vip:             <外部 NW の VIP>
      haas0_op:           <haas0 系の外部 NW の IP>
      haas1_op:           <haas1 系の外部 NW の IP>
      haas2_op:           <haas2 系の外部 NW の IP>
      ext_vip:            192.168.60.15        # Conditional

    network_info:
      mng_subnet:         <内部 NW の subnet>
      mng_netmask:        <内部 NW の netmask>
      ntp_server1:        <外部 NTP サーバ 1 の IP>
      ntp_server2:        <外部 NTP サーバ 2 の IP>
```

```
vip_resource_info:
  mng_vip:
    connection_name:  ens2   ★設定変更不要
    cidr_netmask:     <内部 NW の prefix>
  op_vip:
    connection_name:  ens3   ★設定変更不要
    cidr_netmask:     <外部 NW の prefix>
  ext_vip:
    connection_name:  ens7                # Conditional
    cidr_netmask:     24                  # Conditional
          :               :   ★これ以降は固定値のため変更禁止
```

変数	説明
＜HaaS VM 数＞	HaaS VM の数 「3」を設定
＜IF 数＞	「2」を指定（ネットワーク数）
＜haas0 系のホスト名＞	haas0 系のホスト名
＜haas1 系のホスト名＞	haas1 系のホスト名
＜haas2 系のホスト名＞	haas2 系のホスト名
＜内部 NW の VIP＞	内部通信用 NW の VIP アドレス
＜haas0 系の内部 NW の IP＞	haas0 系の内部通信用 NW の IP アドレス
＜haas1 系の内部 NW の IP＞	haas1 系の内部通信用 NW の IP アドレス
＜haas2 系の内部 NW の IP＞	haas2 系の内部通信用 NW の IP アドレス
＜外部 NW の VIP＞	外部通信用 NW の VIP アドレス
＜haas0 系の外部 NW の IP＞	haas0 系の外部通信用 NW の IP アドレス
＜haas1 系の外部 NW の IP＞	haas1 系の外部通信用 NW の IP アドレス
＜haas2 系の外部 NW の IP＞	haas2 系の外部通信用 NW の IP アドレス
＜内部 NW の subnet＞	内部通信用 NW の subnet
＜内部 NW の netmask＞	内部通信用 NW の netmask
＜外部 NTP サーバ 1 の IP＞	外部 NTP サーバ 1 の IP アドレス
＜外部 NTP サーバ 2 の IP＞	外部 NTP サーバ 2 の IP アドレス
＜内部 NW の prefix＞	内部通信用 NW の prefix
＜外部 NW の prefix＞	外部通信用 NW の prefix

注1: 上記の変数すべてを設定する必要があります。

注2: ホスト名や IP アドレスなどの値の詳細については、『BoxUP 環境定義書_ホスト名』および『BoxUP 環境定義書_IP アドレス』を参照してください。

3-3 IP アドレス書き換え前の確認を行います。

＜入力フォーマット＞

```
# ip a
```

3-4 After config を実行します。

＜入力フォーマット＞

```
# cd /root/after_config
# ./02_construction_change.sh
```

02_construction_change.sh は再起動処理が含まれており、再起動時に自動終了します。

以下は表示例です。

```
# ./02_construction_change.sh
    :（省略）
TASK [reboot all VMs]
*****************************************************************
*******************
Connection to ＜HaaS VM の外部 IP アドレス＞ closed by remote host.
Connection to ＜HaaS VM の外部 IP アドレス＞ closed.
```

注1: TASK [replace /etc/hosts]で止まった場合は「Yes」を入力し続けます。

注2: TASK [Wait for cluster sync (3 node redundancy)]は時間がかかるため、完了するまでしばらく待ちます。

注3: スクリプトを実行すると、「FAILED - RETRYING:～～(100000 retries left).」といった出力が続きますが、エラーではないため問題ありません。そのままお待ちください。

変数	説明
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

3-5 HaaS VM（ACT）にログインします。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

3-6 VIP（冗長 IP アドレス）が変更されていることを確認します。

＜入力フォーマット＞

```
# ip a
```

4 After config 後の正常性確認を実施します。

4-1 正常性確認を行います。

＜入力フォーマット＞

```
# pcs status
```

確認観点は「6.1 クラスタ動作確認」を参照してください。

4-2 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注: ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中の PM、VM から抜けるまで繰り返し実施してください。

5 Zabbix の設定を行います。

5-1 HaaS VM（ACT）にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

5-2 ファイルを修正します。

＜入力フォーマット＞

```
# cd zabbix/
# vi zbx_inventory.ini
```

以下を参考に修正します。

```
[all:vars]
ansible_directory=/srv/zabbix
import_directory=/home/maintenance_tools/zabbix/zabbix_setting/role
s/common_import_items/files
site_region=Unit01-1

default_gateway_ip="＜外部通信用 NW の gateway＞"
haas_vip="haas.vip.mng"
snmp_community="haas_snmp"
           :
           :
```

変数	説明
＜外部通信用 NW の gateway＞	外部通信用 NW の gateway IP アドレス 『BoxUP 環境定義書_IP アドレス』を参照してください。

5-3 Zabbix 設定を実行し、エラーがないことを確認します。

＜入力フォーマット＞

```
# cd /root/zabbix/
# ./setup_zabbix.sh
```

5-4 ホストを確認します。

HaaS PM への SNMP Trap を受信するために、HaaS VM（ACT）で Zabbix のインタフェース設定を修正する手順（STEP 5-4（本手順）～STEP 5-7）を実施します。

＜入力フォーマット＞

```
# ./host_show.sh
```

HaaS PM のホスト名を確認します。

[HOSTNAME]欄に表示されるホスト名のうち、HaaS PM のホスト名を STEP 5-5 のコマンド入力時に使用します。

参考： 「2.1 HaaS ホストの環境構築」で設定したホスト名が HaaS PM のホスト名です。以下は表示例です。

```
[HOSTNAME]                [STATUS] [ZBX]   [SNMP]
HaaSCtlACT                 Enable   Enable  Unknown
haas0                      Enable   Enable  Enable
haas1                      Enable   Enable  Enable
haas2                      Enable   Enable  Enable
haas-host-server-1st       Enable   Enable  Enable
haas-host-server-2nd       Enable   Enable  Enable
haas-host-server-3rd       Enable   Enable  Enable
```

5-5 HaaS PM の設定を確認します。
STEP 5-5~STEP 5-9 は全ての HaaS PM について繰り返し実施してください。

＜入力フォーマット＞

```
# ./host_interface.sh ＜HaaS PM のホスト名＞
```

変数	説明
＜HaaS PM のホスト名＞	STEP 5-4 で表示した HaaS PM のホスト名

リプライの表示例を以下に示します。

```
HOSTNAME : haas-host-server-1st
192.168.80.101  :10050 ZBX  Main
192.168.80.101  :161   SNMP Main
```

「SNMP」の行に表示される IP アドレスによってこの後の手順が異なります。

- 「SNMP」の行に表示される IP アドレスが BMC の IP アドレスでない場合、STEP 5-6 および STEP 5-7 を実施します。
- 「SNMP」の行に表示される IP アドレスが BMC の IP アドレスの場合、STEP 5-8 に進みます。

参考： BMC の IP アドレスを確認するには、該当サーバへログインし以下のコマンドを実施します。

```
# ipmitool lan print | grep "IP Address  "
```

リプライの表示例を以下に示します。

```
# ipmitool lan print | grep "IP Address  "
IP Address              : 192.168.80.246
```

5-6 ホストインタフェースの設定を変更します。

＜入力フォーマット＞

```
# ./hostinterface_update.sh ＜HaaS PM のホスト名＞ ＜BMC の IP アドレス＞
```

変数	説明
＜HaaS PM のホスト名＞	STEP 5-4 で表示した HaaS PM のホスト名
＜BMC の IP アドレス＞	STEP 5-5 の「参考：」で確認した BMC の IP アドレス

コマンドが正常に実行され、「./hostinterface_update.sh end」と表示されることを確認します。

5-7 HaaS PM の設定を確認します。

＜入力フォーマット＞

```
# ./host_interface.sh ＜HaaS PM のホスト名＞
```

変数	説明
＜HaaS PM のホスト名＞	STEP 5-6 で入力したホスト名

リプライの表示例を以下に示します。

```
HOSTNAME : haas-host-server-1st
192.168.80.101  :10050 ZBX  Main
＜BMC の IP アドレス＞  :161   SNMP Main
```

「SNMP」の行に STEP 5-6 で設定した BMC の IP アドレスが表示されることを確認します。

5-8 正常性を確認します。

＜入力フォーマット＞

```
# ./host_show.sh
```

HaaSCtlACT 以外のホスト名すべての[ZBX]および[SNMP]が Enable であることを確認します。SNMP のステータスが Enable にならない場合は STEP 5-9 を実施します。

```
[HOSTNAME]            [STATUS] [ZBX] [SNMP]
HaaSCtlACT            Enable Enable Unknown
haas0                 Enable Enable Enable
haas1                 Enable Enable Enable
haas2                 Enable Enable Enable
haas-host-server-1st  Enable Enable Enable
haas-host-server-2nd Enable Enable Enable
haas-host-server-3rd Enable Enable Enable
```

5-9 SNMP の監視ステータスを更新します。

本手順は 5-8 で SNMP のステータスが Enable とならない場合に実施します。

＜入力フォーマット＞

```
# cd /home/maintenance_tools/zabbix
# ./host_taskcreate.sh ＜ホスト名＞
```

変数	説明
＜ホスト名＞	STEP 5-8 で SNMP が Disable もしくは Unknown となっているホスト名

エラーなく終了することを確認します。また、再度 STEP 5-8 を実施し監視ステータスを確認します。

6 ホスト情報を設定します。

6-1 ホスト情報ファイルを修正します。

＜入力フォーマット＞

```
# vi /srv/common/host_information.yml
```

以下の★印部分を修正します。

```
# vi /srv/common/host_information.yml
senderId: ＜FQDN＞  ★
objectInstance: ＜NF インスタンス名＞  ★
objectClass: HaaS
haas_vip_fqdn: ＜FQDN＞  ★
```

変数	説明
＜FQDN＞	HaaS の外部向け FQDN を設定
＜NF インスタンス名＞	NF インスタンス名を設定

6-2 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注: ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中の PM、VM から抜けるまで繰り返し実施してください。

7 CMDB の HaaS コントローラ情報を登録します。

7-1 HaaS VM（ACT）にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

7-2 CMDB の HaaS コントローラ情報を取得します。

＜入力フォーマット＞

```
# cd /root/config_update_scripts
# ./_haas_ctl_info.sh -mode get
```

現在の HaaS コントローラ情報が表示されます。以下は表示例です。「version」は使用している HaaS コントローラの版数によって変わります。

```
{'vm_name': 'haas0', 'location': '00-01', 'version': '1.5.0'}
{'vm_name': 'haas1', 'location': '00-01', 'version': '1.5.0'}
{'vm_name': 'haas2', 'location': '00-01', 'version': '1.5.0'}
```

7-3 既存の VM 名を指定して、vm_name と location を変更します。

＜入力フォーマット＞

```
# ./_haas_ctl_info.sh -mode set -target ＜変更前の VM 名＞ -vm_name ＜変更後の VM 名＞ -location ＜ラック番号＞
```

変数	説明
＜変更前の VM 名＞	STEP 7-2 で、'vm_name'に表示されたデフォルトの VM 名
＜変更後の VM 名＞	新 VM 名（HaaS VM のホスト名） 『BoxUP 環境定義書_ホスト名』を参照してください。
＜ラック番号＞	ラック番号がデフォルト値となっているため、それぞれの HaaS が搭載されている正しいラック番号を設定します。

7-4 設定が変更されていることを確認します。

以下のコマンドを入力し、STEP 7-3 で設定したとおりに HaaS コントローラの情報が変更されていることを確認します。

＜入力フォーマット＞

```
# ./_haas_ctl_info.sh -mode get
```

8 OAuth2 認証情報を登録します。

8-1 HaaS VM（ACT）にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

8-2 NE-OPS、NW-OPS の OAuth2 認証情報を登録します。

＜入力フォーマット＞

```
# cd /root/config_update_scripts
# ./_generate_oauth2_client.sh ＜Client Name＞
```

変数	説明
＜Client Name＞	NE-OPS もしくは NW-OPS のクライアント名

リプライの表示例を以下に示します。

```
# ./_generate_oauth2_client.sh ne-ops
+ ./scripts/gen_oauth2_client.py --name ne-ops
=== Generate Info ===
Client Name   : ne-ops
Client ID     : 3fca880b-e90a-4373-ae09-cad20ecb19c3
Client Secret :
GYH22ExxoQp3DFH14S0oJ_RJqpOj1qnbrr2v4M0F5dO_idksDKAvCuCr8sq7bZ1S
HyX8cKkZGn6amHY6O3UB8s
```

「Client Name」、「Client ID」、「Client Secret」が出力されることを確認します。

3.5 cron 設定

HaaS VM を構築後、HaaS PM 上で cron 設定を実施します。本手順は全ての HaaS PM に対し実施します。

STEP 操作

1 cron 設定資材のファイルを HaaS PM に転送します。

cron 設定資材ファイルを HaaS PM の「/root」に手動で配置します。

2 HaaS PM にログインします。

HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。

3 cron 設定を行います。

3-1 資材ファイルを解凍します。

＜入力フォーマット＞

```
# cd /root
# tar -xvf if_mon_cronset.tar
# rm -f if_mon_cronset.tar
```

3-2 HaaS PM 上の HaaS VM 名を確認します。

＜入力フォーマット＞

```
# virsh list --all
```

HaaS コントローラの VM 名を確認します。

```
# virsh list --all
Id    名前           状態
-------------------------------
 254   <HaaS VM名>          実行中
```

変数	説明
＜HaaS VM 名＞	HaaS コントローラの VM 名（virsh domain 名）

3-3 資材ファイル内の、サーバに応じて変更が必要な箇所を変更します。

＜入力フォーマット＞

```
# cd /root
# vi cronSetting.cfg
```

以下を参考に変更します。

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin

*/1 * * * * <配置ディレクトリ>/if_mon.sh <HaaS VM名>
```

変数	説明
<配置ディレクトリ>	If_mon.sh を配置したディレクトリ 「/root」に修正してください。
<HaaS VM 名>	STEP 3-2 で確認した HaaS コントローラの VM 名

3-4 cron 設定を反映させます。

<入力フォーマット>

```
# cd /root
# crontab cronSetting.cfg
```

3-5 cron 設定が反映されていることを確認します。

<入力フォーマット>

```
# crontab -l
```

cronSetting.cfg の中身と同じ内容が表示されることを確認します。

3.6 ユーザ作成

一般ユーザの作成を行います。

参考： ユーザ作成の詳細については、『HaaS コントローラ操作マニュアル』の「2.2.1.1　ユーザの管理操作」の「■ユーザの作成」を参照してください。

また、次項にて必要となる ne-ops ユーザの作成を実施します。

以下の手順（STEP 1～STEP 2）は haas0～haas2 に対して実施します。

STEP　操作

1 HaaS VM にログインします。

<入力フォーマット>

```
# ssh <ユーザ名>@<HaaS VM の外部 IP アドレス>
```

変数	説明
<ユーザ名>	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。

変数	説明
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 ne-ops ユーザを作成します。

＜入力フォーマット＞

```
# useradd ne-ops
# passwd ne-ops
password：neops123
```

パスワードは 2 回入力します。

1 回目は、BAD PASSWORD となり、再入力すると以下のように表示されて完了します。

```
          :（省略）
# passwd: すべての認証トークンが正しく更新できました。
```

3 HaaS VM（ACT）にログインします。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

4 /srv/files 配下の所有権を ne-ops ユーザに変更します。

＜入力フォーマット＞

```
# chown -R ne-ops:ne-ops /srv/files
```

3.7 証明書関連設定

証明書に関する設定を行います。

■CA 証明書の設定手順

STEP 操作

1 HaaS VM（ACT）にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注： root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 証明書作成用のディレクトリを作成します。

＜入力フォーマット＞

```
# cd /srv/nginx
# mkdir haas
# cd haas
```

3 秘密鍵を作成します。

以下のコマンドを実行し、任意のパスワードを設定します。

＜入力フォーマット＞

```
# openssl genrsa -aes256 -out ./haas.key 2048
```

以下の表示例を参考に設定します。

```
[root@haasctl00 haas]# openssl genrsa -aes256 -out ./haas.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................+++++
.....................+++++
e is 65537 (0x010001)
Enter pass phrase for ./haas.key:  ★任意のパスワード入力
```

```
Verifying - Enter pass phrase for ./haas.key:  ★パスワードを再入力
```

4 CSR ファイルを作成します。

＜入力フォーマット＞

```
# openssl req -new -key ./haas.key -out haas.csr
```

以下を参考に設定します。

```
[root@haas0 haas]# openssl req -new -key ./haas.key -out haas.csr
Enter pass phrase for ./haas.key:  ★  STEP 3 の「秘密鍵の作成」で入力したパスワードを設定
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:JP  ★「JP」を設定（※以下、CA サーバ環境に合わせて設定）
State or Province Name (full name) []:aaa  ★  ※CA サーバ環境に合わせて設定
Locality Name (eg, city) [Default City]:bbb  ★  ※CA サーバ環境に合わせて設定
Organization Name (eg, company) [Default Company Ltd]:ccc  ★  ※CA サーバ環境に合わせて設定
Organizational Unit Name (eg, section) []:ddd  ★  ※CA サーバ環境に合わせて設定
Common Name (eg, your name or your server's hostname) []:haas  ★  ※HaaS 側の FQDN を設定
Email Address []:  ★未入力で［ENTER］押下

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:  ★未入力で［ENTER］押下
An optional company name []:  ★未入力で［ENTER］押下
```

5 STEP 4 で作成した CSR ファイルを CA サーバへ転送し、CA サーバ上で CA 証明書を発行します。
＜入力フォーマット＞

```
# openssl ca -in <CSR ファイル格納ディレクトリ>/haas.csr -out <証明書発行ディレクトリ>/haas.crt
-days ＜証明書の有効期限＞
```

変数	説明
＜CSR ファイル格納ディレクトリ＞	CSR ファイルを格納したディレクトリ
＜証明書発行ディレクトリ＞	発行した証明書を格納するディレクトリ
＜証明書の有効期限＞	証明書有効期限の日数

コマンドを実行すると入力が必要となるため、以下の表示例を参考に実施します。

```
Enter pass phrase for /home/admin/ca/private/cakey.pem:  ★CA サーバ側のパスワードを入力し Enter 押下

# Sign the certificate? [y/n]:  ★「y」を入力し、Enter 押下

1 out of 1 certificate requests certified, commit? [y/n] ★「y」を入力し、Enter 押下
```

注： CA 証明書の発行で以下のエラーが出た場合、同じ DN の証明書が存在するのが原因です。CA サーバを立てた側において、対象証明書を取り消す必要があります。取り消した証明書は使用不可となるため、対象に注意してください。

```
failed to update database
TXT_DB error number 2
```

6 CA サーバから、以下に示す証明書を HaaS VM の/tmp 配下に転送します。

ファイル	説明	転送先の HaaS VM
HaaS サーバ証明書(haas.crt)	STEP 5 で作成した haas.crt	HaaS VM（ACT）のみ
ルート証明書(NW-OPS)	NW-OPS のルート証明書	HaaS VM（ACT）のみ
ルート証明書(NE-OPS)	NE-OPS のルート証明書	HaaS VM（ACT）のみ
5GC CA 公開鍵	5GC CA の公開鍵	haas.0～haas.2 全て

注： 転送前に CA サーバ上で md5sum コマンドによりチェックサムを取得してください。

7 STEP 6 で転送したファイルをそれぞれ指定のディレクトリにコピーします。

7-1 haas.crt を/srv/nginx/配下にコピーします。

＜入力フォーマット＞

```
# cd /srv/nginx
# rm haas.key haas.crt
# cd /tmp
# cp haas.crt /srv/nginx/
```

7-2 NW-OPS にて使用している CA サーバ上より転送されたルート証明書を/srv/common/配下にコピーします。

＜入力フォーマット＞

```
# cp ＜ルート証明書(NW-OPS)＞ /srv/common/root_cert_nw-ops.pem
```

変数	説明
＜ルート証明書(NW-OPS)＞	NW-OPS にて使用している CA サーバ上より転送されたルート証明書

7-3 NE-OPS にて使用している CA サーバ上より転送されたルート証明書を/srv/common/配下にコピーします。

＜入力フォーマット＞

```
# cp ＜ルート証明書(NE-OPS)＞ /srv/common/root_cert_ne-ops.pem
```

変数	説明
＜ルート証明書(NE-OPS)＞	NE-OPS にて使用している CA サーバ上より転送されたルート証明書

7-4 5GC の CA 公開鍵を/etc/ssh/配下にコピーします。

＜入力フォーマット＞

```
# cp ＜5GC の CA 公開鍵＞ /etc/ssh/
```

変数	説明
＜5GC の CA 公開鍵＞	5GC の CA 公開鍵

7-5 5GC の CA 公開鍵のユーザ所有権を root に変更します。

＜入力フォーマット＞

```
# chown root:root /etc/ssh/＜5GC の CA 公開鍵＞
```

変数	説明
＜5GC の CA 公開鍵＞	5GC の CA 公開鍵

8 チェックサム値を確認します。

＜入力フォーマット＞

```
# cd /srv/nginx
# md5sum haas.crt
# cd /srv/common
# md5sum root_cert_nw-ops.pem
# md5sum root_cert_ne-ops.pem
# cd /etc/ssh
```

```
# md5sum ＜5GC の CA 公開鍵＞
```

変数	説明
＜5GC の CA 公開鍵＞	5GC の CA 公開鍵

チェックサム値が、転送前に CA サーバ上で確認したチェックサム値と同じであることを確認します。

```
# md5sum haas.crt
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx haas.crt

# md5sum root_cert_nw-ops.pem
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx root_cert_nw-ops.pem

# md5sum root_cert_ne-ops.pem
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx root_cert_ne-ops.pem

# md5sum ＜5GC の CA 公開鍵＞
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx ＜5GC の CA 公開鍵＞
```

変数	説明
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	指定されたファイルのチェックサム値（16 進数表記、32 桁）を表示します。
＜5GC の CA 公開鍵＞	5GC の CA 公開鍵

9 NW-OPS のルート証明書と NE-OPS のルート証明書を結合します。

NW-OPS 及び NE-OPS にて使用している各 CA サーバ上より転送されたルート証明書を cacert.pem に追加します。

＜入力フォーマット＞

```
# cd /srv/common
# cat root_cert_nw-ops.pem > cacert.pem
# cat root_cert_ne-ops.pem >> cacert.pem
```

10 STEP 3 で作成した秘密鍵「haas.key」のパスフレーズを解除し、/srv/nginx/にコピーします。

＜入力フォーマット＞

```
# cd /srv/nginx/haas
# cp haas.key haas2.key
# openssl rsa -in haas2.key -out haas.key
# cp haas.key /srv/nginx/
```

```
# cp haas2.key /srv/nginx/
```

openssl コマンドでパスフレーズを解除する際、以下のようにパスワード入力を求められるため、「■CA 証明書の設定手順」の STEP 3 で設定したパスワードを入力し、Enter キーを押下してください。

```
# openssl rsa -in haas2.key -out haas.key
Enter pass phrase for haas2.key:
```

11 設定を有効にするため、系切り替えを実施します。

11-1 HaaS VM のクラスタ設定と ACT ノードを確認します。

＜入力フォーマット＞

```
# pcs status
```

Pacemaker のクラスタ設定が「Online」であることを確認します。

```
# pcs status
    :  (省略)
Node List:
  * Online: [ haas.0 haas.1 haas.2 ]

Full List of Resources:
  * Resource Group: haas:
    * filesystem       (ocf::heartbeat:Filesystem):    Started haas.0
```

注: 「Started」と表示されているノードが ACT ノードです。

11-2 STEP 11-1 で確認した ACT ノード以外の HaaS VM にログインします。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 IP アドレス＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注: root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

11-3 ACT 以外のノード上で系切り替えを行います。

＜入力フォーマット＞

```
# pcs node standby ＜ACT のノード名＞
```

変数	説明
＜ACT のノード名＞	HaaS VM（ACT）のノード名

11-4 系切り替え完了後、旧 ACT ノードを standby から復帰させます。

＜入力フォーマット＞

```
# pcs node unstandby ＜ノード名＞
```

変数	説明
＜ノード名＞	STEP 11-3 で指定した HaaS VM（旧 ACT）のノード名

11-5 Pacemaker のクラスタ設定を確認します。

＜入力フォーマット＞

```
# pcs status
```

Pacemaker のクラスタ設定が「Online」であることを確認します。

```
# pcs status
    :  (省略)
Node List:
  * Online: [ haas.0 haas.1 haas.2 ]
```

12 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注： ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中の PM、VM から抜けるまで繰り返し実施してください。

■NE-OPS→HaaS の ssh/sftp 通信設定

以下の手順（STEP 1～STEP5）を haas0～haas2 に対して実施します。

STEP	操作

1 HaaS VM にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 IP アドレス＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 ssh ディレクトリを作成します。

2-1 /home/ne-ops に ssh ディレクトリを作成します。

＜入力フォーマット＞

```
# cd /home/ne-ops
# mkdir .ssh
```

2-2 ディレクトリの権限を設定します。

＜入力フォーマット＞

```
# chmod 700 .ssh
```

3 sshd_config を修正します。

＜入力フォーマット＞

```
# cd /etc/ssh
# vi sshd_config
```

以下を参考に「TrustedUserCAKeys」の設定を修正します（★印部分）。

```
          :（省略）
# BEGIN ANSIBLE MANAGED BLOCK EOF
TrustedUserCAKeys /etc/ssh/＜5GC の CA 公開鍵＞ ★
# END ANSIBLE MANAGED BLOCK EOF
```

変数	説明
＜5GC の CA 公開鍵＞	「■CA 証明書の設定手順」の STEP7-4 で/etc/ssh 配下に格納した 5GC の CA 公開鍵

4 ssh サービスをリスタートします。

＜入力フォーマット＞

```
# systemctl restart sshd.service
# systemctl status sshd.service
```

エラーがなく、「Active: active (running)」と表示されることを確認します。

```
# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor
preset: enabled)
   Active: active (running) since Tue 2021-11-16 22:56:34 JST; 1min 37s
ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 40244 (sshd)
    Tasks: 1 (limit: 410379)
   Memory: 1.2M
   CGroup: /system.slice/sshd.service
           mq40244 /usr/sbin/sshd -D -oCiphers=aes256-
gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-
cbc,aes128-gcm@openssh.com,aes128-ctr>
:
```

5 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注： ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中の PM、VM から抜けるまで繰り返し実施してください。

■証明書の導通確認

STEP　操作

1 HaaS VM（ACT）にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名 初期構築時は operator ユーザを使用してください。 パスワードは「opepass」です。
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注： root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 証明書の接続を確認します。

クライアント：HaaS　サーバ：NW-OPS、NE-OPS としての接続を確認します。

＜入力フォーマット＞

```
# openssl s_client -CAfile /srv/common/cacert.pem -connect ＜NW-OPS の FQDN＞:10443
# openssl s_client -CAfile /srv/common/cacert.pem -connect ＜NW-OPS の FQDN＞:9443
# openssl s_client -CAfile /srv/common/cacert.pem -connect ＜NE-OPS の FQDN＞:443
```

変数	説明
＜NW-OPS の FQDN＞	NW-OPS の FQDN を指定
＜NE-OPS の FQDN＞	NE-OPS の FQDN を指定

エラーが出ていないことを確認します。

注： エラーが発生した場合、『HaaS コントローラ操作マニュアル』の「3.2.1　外部装置との疎通確認」を参照し、NW-OPS、NE-OPS への疎通に問題が無い事を確認して下さい。

3.8 事後作業

不要なファイルの削除を実施します。

本手順は全ての HaaS PM に対して実施します。(順不同)

STEP	操作
1	HaaS PM にログインします。 HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。
2	不要な圧縮 VM イメージファイルを削除します。 以下のコマンドを実行します。 ＜入力フォーマット＞

```
# cd /home/images
# rm <圧縮イメージファイル名>
```

変数	説明
＜圧縮イメージファイル名＞	圧縮 VM イメージファイル名 例：haasctl_xxxx_comp.qcow2

3 不要な XML ファイルを削除します。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# cd /tmp
# rm ＜XML ファイル名＞
```

変数	説明
＜XML ファイル名＞	XML ファイル名 例：0 系の場合　haasctl00_xxxx.xml 1 系の場合　haasctl01_xxxx.xml 2 系の場合　haasctl02_xxxx.xml

4 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注： ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中 PM から抜けるまで繰り返し実施してください。

4 各種設定

4.1 ログ収集設定

HaaS コントローラを起動した時点でログ収集の設定は完了しているため、追加で必要な設定はありません。

ログ収集の設定内容を確認する手順を以下に示します。

STEP 操作

1 HaaS VM にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 IP アドレス＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注： root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 ログ収集の設定内容を確認します。

操作ログの取得設定を確認する場合

＜入力フォーマット＞

```
# cat /root/.bash_profile
```

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
PATH=$PATH:$HOME/bin
export PATH
if [ $(whoami) = "root" ]; then
        log_archive_directory=/var/log/operation
else
        log_archive_directory=/home/$(whoami)/log
```

```
fi

script -f ${log_archive_directory}/$(date +%Y%m%d_%H%M%S)_$(whoami).log
```

アプリログ（docker ログ）の取得設定を確認する場合

＜入力フォーマット＞

```
# cat /etc/rsyslog.d/10-docker.conf
```

```
$template DockerLogs,
"/var/log/docker/%programname%_%$year%%$month%%$day%.log"

if $syslogfacility-text == 'daemon' and $programname contains 'docker-
apps' then -?DockerLogs
& stop
```

5 インストールファイルとサーバ設定の確認

5.1 OS バージョン／カーネルバージョン情報の確認

5.1.1 PM OS バージョン／カーネルバージョン情報の確認

HaaS PM を対象に以下の手順を実施します。

STEP	操作

1 HaaS PM にログインします。

HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。

2 OS バージョン情報を確認します。

＜入力フォーマット＞

```
# cat /etc/redhat-release
```

OS バージョンが表示されることを確認します。

```
# cat /etc/redhat-release

CentOS Linux release 8.2.2004 (Core)
```

3 カーネルバージョン情報を確認します。

＜入力フォーマット＞

```
# uname -a
```

「4.18.0-193.14.2.el8_2.ng101.x86_64」が表示されることを確認します。

```
# uname -a

Linux ＜PM のホスト名＞4.18.0-193.14.2.el8_2.ng101.x86_64 #1 SMP Sun Jul 26
03:54:29 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

4 ログアウトします。

＜入力フォーマット＞

```
# exit
```

5.1.2 VM OS バージョン／カーネルバージョン情報の確認

HaaS VM を対象に以下の手順を実施します。

STEP 操作

1 HaaS VM にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 IP アドレス＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注： root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 OS バージョン情報を確認します。

＜入力フォーマット＞

```
# cat /etc/redhat-release
```

OS バージョンが表示されることを確認します。

```
# cat /etc/redhat-release

CentOS Linux release 8.2.2004 (Core)
```

3 カーネルバージョン情報を確認します。

＜入力フォーマット＞

```
# uname -a
```

「4.18.0-193.14.2.el8_2.ng102.x86_64」が表示されることを確認します。

```
# uname -a

Linux ＜VM のホスト名＞ 4.18.0-193.14.2.el8_2.ng102.x86_64 #1 SMP Sun Jul 26
03:54:29 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

4 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注： ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中の PM、VM から抜けるまで繰り返し実施してください。

5.2 サーバ設定確認

5.2.1 PM IP アドレス／サブネットマスクの確認

HaaS PM を対象に以下の手順を実施します。

STEP	操作

1 HaaS PM にログインします。

HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。

2 IP アドレスとサブネットマスク設定を確認します。

＜入力フォーマット＞

```
# ip a
```

『BoxUP 環境定義書_IP アドレス』の HaaS PM のアドレスとサブネットマスクが「br-haas-op」および「br-haas-mng」のインタフェースに設定されていることを確認します。以下は表示例です。

```
# ip a

          : （省略）
9: br-haas-mng: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
group default qlen 1000
    link/ether 90:e2:ba:39:45:25 brd ff:ff:ff:ff:ff:ff
    inet <内部 IP アドレス> / <プレフィックス> brd 192.168.70.255 scope global
noprefixroute br-haas-mng
       valid_lft forever preferred_lft forever
    inet6 fe80::8cb9:497e:6923:6fec/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
10: br-haas-op: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
group default qlen 1000
    link/ether 80:30:e0:1d:a7:f0 brd ff:ff:ff:ff:ff:ff
    inet <外部 IP アドレス> / <プレフィックス> brd 192.168.80.255 scope global
noprefixroute br-haas-op
       valid_lft forever preferred_lft forever
    inet6 fe80::7e10:541f:6712:e84a/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
    link/ether fe:54:00:66:06:7f brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fc54:ff:fe66:67f/64 scope link
       valid_lft forever preferred_lft forever
          : （省略）
```

3 ログアウトします。

＜入力フォーマット＞

```
# exit
```

5.2.2 VM IP アドレス／サブネットマスクの確認

HaaS VM を対象に以下の手順を実施します。

STEP 操作

1 HaaS VM にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 IP アドレス＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注： root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 IP アドレスとサブネットマスク設定を確認します。

＜入力フォーマット＞

```
# ip a
```

『BoxUP 環境定義書_IP アドレス』の各 Haas VM のアドレスとサブネットマスクが「ens2」および「ens3」のインタフェースに設定されていることを確認します。

```
# ip a

          : （省略）
2: ens2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 52:54:00:4d:04:9c brd ff:ff:ff:ff:ff:ff
    inet <内部 IP アドレス> / <プレフィックス> brd 192.168.80.255 scope global
noprefixroute ens2
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe4d:49c/64 scope link
       valid_lft forever preferred_lft forever
3: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 52:54:00:28:79:35 brd ff:ff:ff:ff:ff:ff
```

```
   inet <外部 IP アドレス> / <プレフィックス> brd 192.168.70.255 scope global
noprefixroute ens3
      valid_lft forever preferred_lft forever
         : （省略）
```

3 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注： ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中の PM、VM から抜けるまで繰り返し実施してください。

5.2.3 PM NTP の確認

HaaS PM を対象に以下の手順を実施します。

STEP 操作

1 HaaS PM にログインします。

HaaS PM へのログイン手順については、「2 PM の設定」の「■HaaS PM へのログイン」を参照してください。

2 NTP 設定を確認します。

＜入力フォーマット＞

```
# chronyc sources
```

接続先（Number of sources）が 2 つで、S 列に「*」と「+」または「*」と「-」が表示されることを確認します。

また、NTP サーバの IP アドレスが環境定義書と一致していることを確認します。

以下は、「*」と「+」の場合での表示例となります。

```
# chronyc sources

210 Number of sources = 2
MS Name/IP address    Stratum  Poll  Reach  LastRx  Last sample
===============================================================================
^* <NTP サーバ 1>              1    6     17     26   -38us[ -112us]+/- 41ms
^+ <NTP サーバ 2>              1    6     17     26   -69us[ -113us]+/- 41ms
```

変数	説明
＜NTP サーバ 1＞、＜NTP サーバ 2＞	NTP サーバ 1 および NTP サーバ 2 の IP アドレス

3 ログアウトします。

＜入力フォーマット＞

```
# exit
```

5.2.4 VM NTP の確認

HaaS VM を対象に以下の手順を実施します。

STEP 操作

1 HaaS VM にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 IP アドレス＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名
＜HaaS VM の外部 IP アドレス＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注： root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 NTP 設定を確認します。

＜入力フォーマット＞

```
# chronyc sources
```

接続先（Number of sources）が 2 つで、S 列に「*」と「+」または「*」と「-」が表示されることを確認します。

また、NTP サーバの IP アドレスが環境定義書と一致していることを確認します。

以下は、「*」と「+」の場合での表示例となります。

```
# chronyc sources

210 Number of sources = 2
MS Name/IP address    Stratum  Poll  Reach  LastRx  Last sample
===============================================================================
^* ＜NTP サーバ 1＞              1    6     17     26   -38us[ -112us]+/- 41ms
^+ ＜NTP サーバ 2＞              1    6     17     26   -69us[ -113us]+/- 41ms
```

変数	説明
＜NTP サーバ 1＞、＜NTP サーバ 2＞	NTP サーバ 1 および NTP サーバ 2 の IP アドレス

3 ログアウトします。

＜入力フォーマット＞

```
# exit
```

注： ログスクリプトの停止や root 権限、一般ユーザのログアウトのため複数回実施する必要があります。ログイン中の PM、VM から抜けるまで繰り返し実施してください。

5.3 ミドルウェアバージョン確認

本項の 5.3.1～5.3.13 に示す各手順では、はじめに HaaS コントローラの ACT VM へのログインが必要です。以下にログイン手順を示します。

STEP	操作
1	ACT VM にログインします。 以下のコマンドを実行します。 ＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注： root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

5.3.1 Pacemaker バージョン確認

Pacemaker のバージョン確認手順を以下に示します。

STEP	操作
1	バージョン情報を確認します。 ＜入力フォーマット＞

```
# pacemakerd --version
```

「2.0.4-6」であることを確認します。

```
Pacemaker 2.0.4-6.el8_3.1
Written by Andrew Beekhof
```

5.3.2 Corosync バージョン確認

Corosync のバージョン確認手順を以下に示します。

STEP	操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# corosync -v
```

「3.0.3」であることを確認します。

```
Corosync Cluster Engine, version '3.0.3'
Copyright (c) 2006-2018 Red Hat, Inc.
```

5.3.3 DRBD バージョン確認

DRBD のバージョン確認手順を以下に示します。

STEP	操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# drbdadm --version
```

「9.13.1」であることを確認します。

```
DRBDADM_BUILDTAG=GIT-
hash:¥ b24b0f7e42d500d3538d7eeffa017ec78d08f918¥ build¥ by¥ mockbuild@¥,¥
2020-06-24¥ 03:20:32
DRBDADM_API_VERSION=2
DRBD_KERNEL_VERSION_CODE=0x090017
DRBD_KERNEL_VERSION=9.0.23
DRBDADM_VERSION_CODE=0x090d01
DRBDADM_VERSION=9.13.1
```

5.3.4 Cobbler バージョン確認

Cobbler のバージョン確認手順を以下に示します。

STEP	操作

1 **バージョン情報を確認します。**

＜入力フォーマット＞

```
# cobbler --version
```

「3.1.2」であることを確認します。

```
# cobbler --version

Cobbler 3.1.2
  source: ?, ?
  build time: Sun May 31 02:32:34 2020
```

5.3.5 Docker バージョン確認

Docker のバージョン確認手順を以下に示します。

STEP	操作

1 **バージョン情報を確認します。**

＜入力フォーマット＞

```
# docker --version
```

「19.03.13」であることを確認します。

```
Docker version 19.03.13, build 48a66213fe
```

5.3.6 AWX バージョン確認

AWX のバージョン確認手順を以下に示します。

STEP	操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# docker exec -it awx_task /bin/bash
bash-4.4# awx --version
```

「14.0.0」であることを確認します。

```
14.0.0
```

2 docker コンテナから抜けます。

「Ctrl」キーと「p」キーを同時に押し、次に「Ctrl」キーと「q」キーを同時に押します。

5.3.7 ansible バージョン確認

ansible のバージョン確認手順を以下に示します。

STEP	操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# docker exec -it awx_task /bin/bash
bash-4.4# ansible --version
```

「2.9.11」であることを確認します。

```
ansible 2.9.11
  config file = /etc/ansible/ansible.cfg
  configured module search path =
['/var/lib/awx/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.6/site-
packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.6.8 (default, Apr 16 2020, 01:36:27) [GCC 8.3.1
20191121 (Red Hat 8.3.1-5)]
```

2 docker コンテナから抜けます。

「Ctrl」キーと「p」キーを同時に押し、次に「Ctrl」キーと「q」キーを同時に押します。

5.3.8 Zabbix サーバ バージョン確認

Zabbix サーバのバージョン確認手順を以下に示します。

STEP	操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# docker exec -it zabbix_server /bin/bash
[root@abd0fc6fc62e zabbix]# zabbix_server --version
```

「4.0.22」であることを確認します。

```
zabbix_server (Zabbix) 4.0.22
Revision 073cb9f 29 June 2020, compilation time: Jun 29 2020 13:25:09

Copyright (C) 2020 Zabbix SIA
License GPLv2+: GNU GPL version 2 or later
<http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it
according to
the license. There is NO WARRANTY, to the extent permitted by law.

This product includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit (http://www.openssl.org/).

Compiled with OpenSSL 1.0.2k-fips  26 Jan 2017
Running with OpenSSL 1.0.2k-fips  26 Jan 2017
```

2 docker コンテナから抜けます。

「Ctrl」キーと「p」キーを同時に押し、次に「Ctrl」キーと「q」キーを同時に押します。

5.3.9 Zabbix Agent バージョン確認

Zabbix Agent のバージョン確認手順を以下に示します。

STEP	操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# zabbix_agentd --version
```

「4.0.22」であることを確認します。

```
zabbix_agentd (daemon) (Zabbix) 4.0.22
Revision 073cb9f 29 June 2020, compilation time: Jun 29 2020 14:13:44

Copyright (C) 2020 Zabbix SIA
License GPLv2+: GNU GPL version 2 or later
<http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it
according to
the license. There is NO WARRANTY, to the extent permitted by law.

This product includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit (http://www.openssl.org/).

Compiled with OpenSSL 1.0.2k-fips  26 Jan 2017
Running with OpenSSL 1.0.2k-fips  26 Jan 2017
```

5.3.10 GitLab バージョン確認

GitLab のバージョン確認手順を以下に示します。

STEP	操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# docker exec -it gitlab /bin/bash
root@402953f88829:/# gitlab-rake gitlab:env:info
```

「13.2.4」であることを確認します。

```
            : （省略）
GitLab information
Version:       13.2.4
Revision:      136d3a02dca
Directory:     /opt/gitlab/embedded/service/gitlab-rails
DB Adapter:    PostgreSQL
DB Version:    11.7
URL:           http://192.168.80.130/gitlab
HTTP Clone URL: http://192.168.80.130/gitlab/some-group/some-project.git
SSH Clone URL:  git@192.168.80.130:some-group/some-project.git
Using LDAP:    no
Using Omniauth: yes
Omniauth Providers:
            : （省略）
```

2 docker コンテナから抜けます。

「Ctrl」キーと「p」キーを同時に押し、次に「Ctrl」キーと「q」キーを同時に押します。

5.3.11 PostgreSQL バージョン確認

PostgresSQL のバージョン確認手順を以下に示します。

STEP	操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# docker exec -it postgres /bin/bash
root@952a95deed0c:/# postgres --version
```

「10.14」であることを確認します。

```
postgres (PostgreSQL) 10.14
```

2 docker コンテナから抜けます。

「Ctrl」キーと「p」キーを同時に押し、次に「Ctrl」キーと「q」キーを同時に押します。

5.3.12 Nginx バージョン確認

Nginx のバージョン確認手順を以下に示します。

STEP 操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# docker exec -it nginx /bin/bash
root@9c2d67144c92:/# nginx -v
```

「1.19.1」であることを確認します。

```
nginx version: nginx/1.19.1
```

2 docker コンテナから抜けます。

「Ctrl」キーと「p」キーを同時に押し、次に「Ctrl」キーと「q」キーを同時に押します。

5.3.13 Django バージョン確認

Django のバージョン確認手順を以下に示します。

STEP 操作

1 バージョン情報を確認します。

＜入力フォーマット＞

```
# docker exec -it cmdb /bin/bash
root@eb240a3ff12b:/django# django-admin --version
```

「3.1」であることを確認します。

```
3.1
```

2 docker コンテナから抜けます。

「Ctrl」キーと「p」キーを同時に押し、次に「Ctrl」キーと「q」キーを同時に押します。

6 システム動作の確認

6.1 クラスタ動作確認

Pacemaker および Corosync によるクラスタ状態の正常性を確認します。

STEP 操作

1 HaaS VM（ACT）にログインします。

以下のコマンドを実行します。

＜入力フォーマット＞

```
# ssh ＜ユーザ名＞@＜HaaS VM の外部 VIP＞
```

変数	説明
＜ユーザ名＞	一般ユーザのアカウント名
＜HaaS VM の外部 VIP＞	『BoxUP 環境定義書_IP アドレス』の外部接続を参照してください。

注： root 権限を使った操作をする為、一般ユーザからのログイン後、su - で root 権限に切り替えてください。

2 ACT VM 上でクラスタの正常性を確認します。

＜入力フォーマット＞

```
# pcs status
```

「Node List」にクラスタ構成を組んでいるノード全てが「Online」となっていること、「Full list of resource」に表示されるリソースの状態が「Started」と表示されていることを確認します。

```
# pcs status
        :  (省略)
Node List:
  * Online: [ ccc...c ]
        :  (省略)
Full list of resources:

  * ＜リソース名＞:        Started ＜ノード名＞
  * ＜リソース名＞:        Started ＜ノード名＞
        :  (省略)
```

変数	説明
Online: [ccc...c]	Online 状態のノード名を半角スペースで区切って表示します。 表示例：haas.0 haas.1 haas.2 **注：** Online 状態のノードがない場合は、項目自体を表示しません。
＜リソース名＞	filesystem、postgres などの個別のリソース名を表示します。
＜ノード名＞	個別のノード名を表示します。 表示例：haas.0

6.2 外部装置との疎通確認

HaaS コントローラと外部装置との間の疎通を確認します。

参考： 外部装置との疎通確認の詳細については、『HaaS コントローラ操作マニュアル』の「3.2.1 外部装置との疎通確認」を参照してください。

導入マニュアル

2022年11月　　第5.4版

NEC Corporation