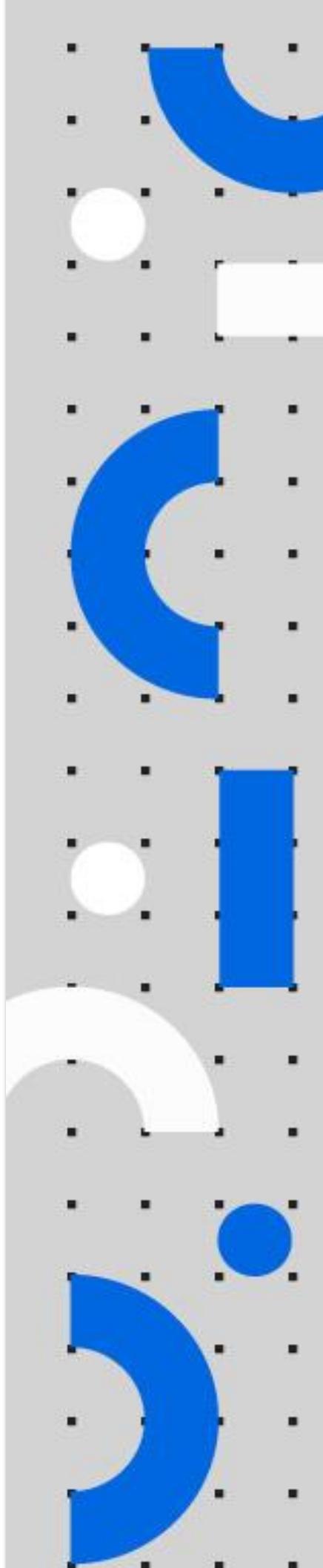


# **UiPath High Availability Add-on**

## インストールガイド

**v2023.4 対応版**

Information contained herein is confidential and may be disclosed only with the written permission of UiPath.  
All rights reserved.



## リビジョン履歴

Date	Version	Author	Description
27 <sup>th</sup> July 2020	1.0	UiPath Infra Japan Team	First version for v2020.4 release
4 <sup>th</sup> Aug 2020	1.1	UiPath Infra Japan Team	Updated for v2.0 script
12 <sup>th</sup> June 2023	2.0	UiPath Infra Japan Team	Updated for v2023.4 release

## 商標について

- UiPath のソフトウェア、製品、サービス、(これには、UiPath Orchestrator、UiPath Robot、UiPath Studio が含まれますが、これらに限りません)はアメリカ合衆国で登録された UiPath Inc.、および他の国・地域で登録された UiPath の関係会社の商標または登録商標です。UiPath のロゴは UiPath Inc. が所有するものであり、UiPath の事前の明示的な許可なく、お客様及びその他の方が使用することはできません。
- Microsoft のソフトウェア、製品、サービス(これには、Microsoft、Windows、Windows Server、SQL Server 及び Active Directory が含まれますが、これらに限りません)はアメリカ合衆国で登録された Microsoft Corporation 及び他の国・地域で登録されたその関係会社の商標または登録商標です。
- Oracle のソフトウェア、製品、サービス(これには、Java も含まれますがこれに限りません)はアメリカ合衆国で登録された Oracle 及びその他の国・地域で登録された関係会社の商標または登録商標です。
- Elastic は、Elastic N.V. 及びその関係会社の商標または登録商標です。
- Redis は、Redis Labs Ltd の商標です。
- その他、記載されている製品名、会社名およびサービス名はそれぞれの各社の商標または登録商標です。

## 免責事項

- 本ガイドの内容は 2023 年 5 月現在の情報であり、本ガイドは、下記の製品（以下「本製品」といいます。）を説明するものです。
  - UiPath High Availability Add-on
- 本製品、本製品を構成するプログラムまたは本製品が依存しもしくは本製品と連携する外部サービスがアップデートまたは修正される等によって、本製品が本ガイドの説明と異なる動作をする、または仕様となる可能性がありますので、予めご留意ください。当該アップデートまたは修正等の後に本ガイドが改訂された場合には、当該改訂後の本ガイドが本製品を説明するものとなります。
- 本ガイドに含まれる情報は可能な限り正確を期しておりますが、本ガイドに記載された内容の正確性、充分性等に関して UiPath 株式会社（以下「UiPath」といいます。）は何ら保証していません。従って、本ガイドに含まれる情報の利用はお客様の責任においてなされるものであり、UiPath はガイドの内容によってお客様が受けたいかなる損害に関して何らの補償をするものではありません。
- 本ガイドは UiPath を法的に拘束する書類ではありません。UiPath はお客様に通知なくして、本ガイドの内容の一部または全部を修正およびアップデートできます。
- 本ガイドの著作権などの一切の知的財産権は、UiPath またはその関係会社に帰属します。お客様は UiPath の書面の承諾なしで本ガイドを複製、修正、頒布、公衆送信等できません。

## 目次

1.	はじめに .....	5
1.1.	本文書の目的と対象読者 .....	5
1.2.	HAA の役割 .....	5
1.3.	HAA 構成例 .....	6
2.	インストール前準備.....	8
2.1.	システム要件とサーバー準備.....	8
2.2.	DNS サーバー利用要否の検討 .....	8
3.	HAA インストール手順.....	9
3.1.	HAA インストール前準備 .....	9
3.2.	1 台目の HAA インストール .....	11
3.3.	2 台目以降の HAA インストール.....	12
3.4.	オフライン環境での HAA インストール .....	13
3.5.	DNS サーバーの設定 .....	17
3.6.	HAA 管理コンソールでの操作.....	20
3.7.	HAA 管理コンソールでの冗長構成確認 .....	22
3.8.	HAA アンインストール .....	25
4.	Orchestrator の HAA 接続設定 .....	26
5.	デバッグ情報収集 .....	28
6.	技術支援のご案内 .....	29

# 1. はじめに

## 1.1. 本文書の目的と対象読者

- 本文書では UiPath High Availability Add-on (以降 HAA と略す) のインストール手順について説明します。
- 対象読者には以下の知識を有していることを前提としています。
  - 利用するオペレーティングシステム(Windows Server, Linux)の操作方法についての一般的な知識
  - TCP/IP ネットワークについての一般的な知識
  - UiPath Orchestrator の概要、インストールおよび基本操作の知識

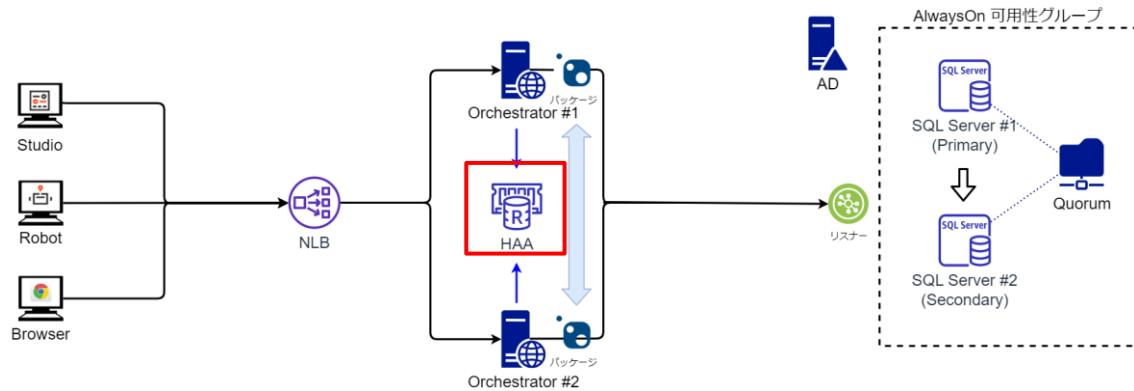
## 1.2. HAA の役割

- UiPath Orchestrator を Active-Active な冗長構成にする場合には、HAA または Redis のサーバーが必須となります。
- HAA (Redis) はインメモリ DB として動作し、セッション情報の管理、および DB 情報を一部キャッシングし、パフォーマンス向上に寄与するという 2 つの役割があります。
- HAA (Redis) そのものを冗長化させることも可能で、冗長構成では Primary から Replica にリアルタイムで同期されるため、DB 内の情報を永続化する必要はありません。HAA のシステム要件と役割については次のサイトもご参照ください。
  - <https://docs.uipath.com/ja/orchestrator/standalone/2023.4/installation-guide/haa-hardware-and-software-requirements>
- Redis オープンソース版またはパブリッククラウドにおける Redis マネージドサービスと Orchestrator を連携させることは技術的には可能ですが、Redis 本体は UiPath 社ではサポート対象外となります。
- HAA は、Redis 社が提供する Redis オープンソースをベースとした商用版製品である Redis Enterprise を UiPath 社が OEM で提供する製品です。HAA のライセンスをご購入のお客様には UiPath 社がテクニカルサポートを提供しております。Redis 本体を含めた包括的なサポートが必要な場合には、HAA の導入を是非ご検討ください。
- HAA はトライアルライセンスの申込不要で 30 日間無償で試用することができます。ライセンスをご購入いただき、トライアルにて構築した環境をそのまま本番利用することも可能です。まずはこのガイドを元に実証検証していただくことを推奨いたします。ただしトライアル段階ではテクニカルサポートサービスは提供できない旨、ご了承ください。
- なお HAA を含めた Orchestrator の全体的な設計については「UiPath Orchestrator システムの基盤設計・運用ガイド」をご参照ください。
  - <https://www.uipath.com/ja/community-blog/knowledge-base/uipath-orchestrator-design-operation-guide>

### 1.3. HAA 構成例

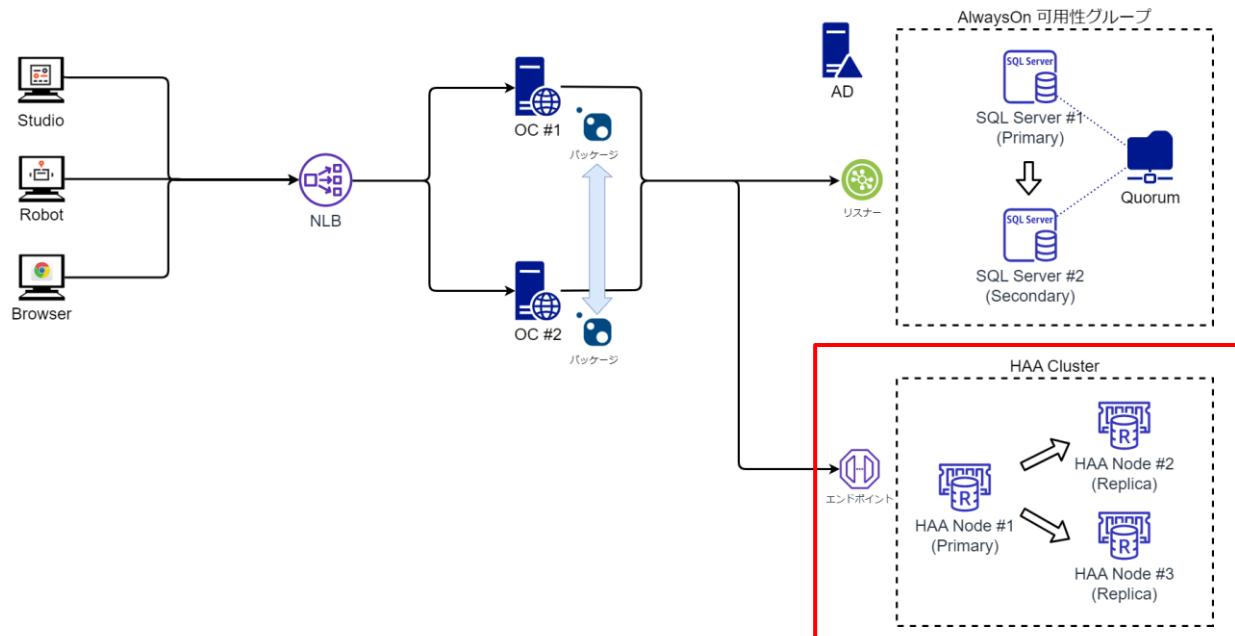
- 一部冗長化、完全冗長化および地理的冗長化の構成を例示します。

➤ 一部冗長化



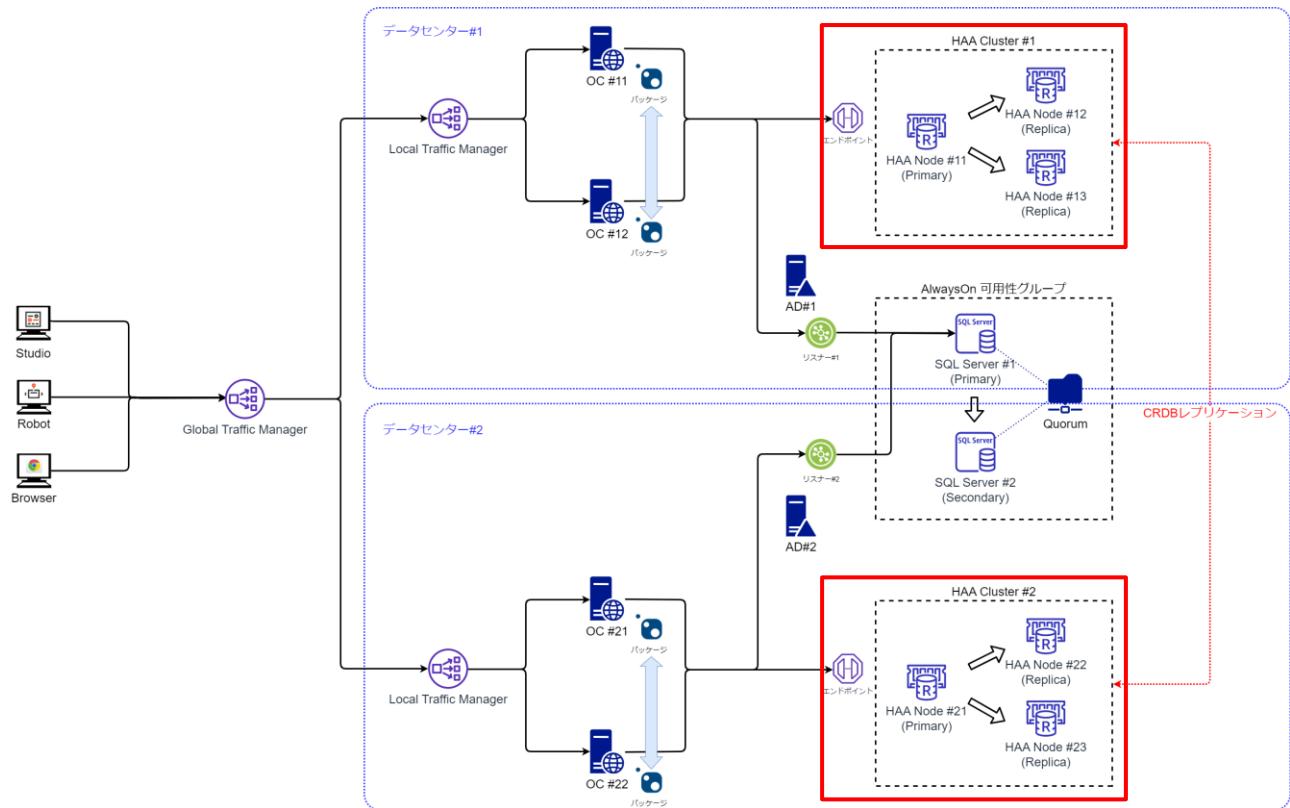
✧ この構成のように HAA は 1 台で運用することができますが単一障害点となります。

➤ 完全冗長化



✧ 可用性を確保するには完全冗長化構成のように HAA を 3 台ノードで冗長構成にします。  
✧ HAA ライセンスはクラスタ単位となり、ノードが 1 台でも 3 台でも費用は同じです。

➤ 地理的冗長化 (ディザスタリカバリー構成)



- ❖ 2つのデータセンターにそれぞれ Orchestrator 環境を構築し、ディザスタリカバリー (DR) 構成を取ることも可能です。
- ❖ HAA はデータセンターごとにクラスタを組み、両者は CRDB (Conflict-free Replicated Database) レプリケーションを使用して整合性を保ちつつデータベースの同期を行うことができます。
- ❖ HAA ライセンスはクラスタ単位となりますので、この構成では 2つの HAA ライセンスが必要となることに注意してください。
- ❖ 本文書では DR 構成のセットアップ手順については取り扱いませんので、詳細については次のガイドをご参照ください: <https://docs.uipath.com/ja/orchestrator/standalone/2023.4/installation-guide/haa-disaster-recovery-activeactive-installation>

## 2. インストール前準備

### 2.1. システム要件とサーバー準備

- 次の Web ガイドを参照しシステム要件を満たすサーバーマシンを準備します。パブリッククラウドのインスタンスを使用することも可能です。
  - <https://docs.uipath.com/ja/orchestrator/standalone/2023.4/installation-guide/haa-hardware-and-software-requirements>
- 本文書では HAA をインストールするホストとして Red Hat Enterprise Linux (RHEL) 8.6、SSH クライアントは [Tera Term](#) 4.106 を使用しています。
  - RHEL (Red Hat Enterprise Linux) においても手順は同じになります。
  - SSH クライアントまたはコマンドプロンプト上で実行するコマンドは下記のようにグレーの背景で表記します。

```
chmod +x get-haa.sh
```

- HAA インストールには下記のユーザーおよび権限が必要となります。
  - root、または sudo でコマンド実行可能なユーザー
  - ユーザー作成権限
- 接続安定化のため固定 IP アドレスの使用を推奨します。
- HAA が使用するポートについては下記 Web サイトをご参照ください。HAA インストーラーが必要なファイアウォールポートを自動的に開放します。
  - <https://docs.uipath.com/ja/orchestrator/standalone/2023.4/installation-guide/haa-hardware-and-software-requirements#ports>
- 環境に応じて手順が分岐する部分は 【HAA 冗長構成の場合】 のように 【】 付きで示しています。
- HAA インストールの手順については Web ガイドもご参照ください。
  - <https://docs.uipath.com/ja/orchestrator/standalone/2023.4/installation-guide/haa-installation>

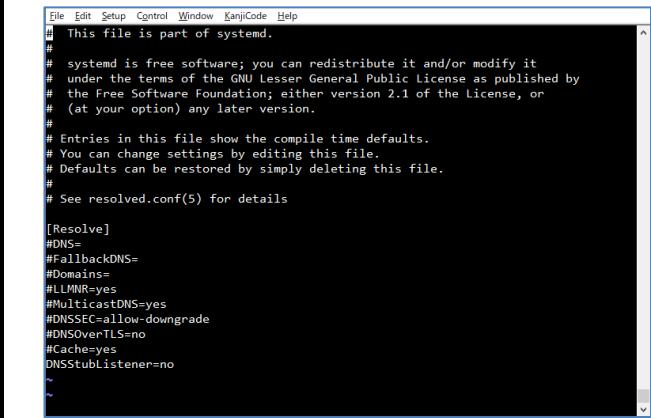
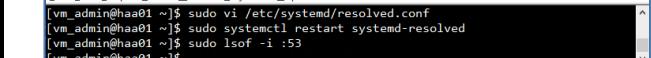
### 2.2. DNS サーバー利用要否の検討

- HAA 冗長構成では Orchestrator が参照する DNS サーバーの NS レコードを使用して、HAA 接続のエンドポイントを個別の IP アドレスではなく、たとえば redis-10000.haa-cluster.lab.test:10000 のように一つに集約することができます。このエンドポイントは現在マスターとなっている HAA ノードに自動的に名前解決されます。
- DNS サーバーを利用する場合の作業手順は下記の通りとなります。
  - HAA クラスタの FQDN を策定します。本文書では **haa-cluster.lab.test** とします。
  - [HAA インストール前準備](#)、[1台目の HAA インストール](#)、[2台目以降の HAA インストール](#) をそれぞれ参照して、3台のノードの HAA をインストールします。
  - [DNS サーバーの設定](#)を参照して DNS サーバーの設定を行います。

## 3. HAA インストール手順

### 3.1. HAA インストール前準備

- HAA をインストールする前準備の手順について説明します。冗長構成ではすべてのノードに対して本節の手順を実行します。
- 以降、HAA ホストに SSH クライアントでアクセスした上で実行します。

 <pre>File Edit Setup Control Window KanjiCode Help # This file is part of systemd. # # systemd is free software; you can redistribute it and/or modify it # under the terms of the GNU Lesser General Public License as published by # the Free Software Foundation; either version 2.1 of the License, or # (at your option) any later version. # # Entries in this file show the compile time defaults. # You can change settings by editing this file. # Defaults can be restored by simply deleting this file. # # See resolved.conf(5) for details  [Resolve] #DNS= #FallbackDNS= #Domains= #LLMNR=yes #MulticastDNS=yes #ONSECSECallow-downgrade #DNSOverTLS=no #Cache=yes DNSStubListener=no ~ ~</pre>	<p>HAA は内部的に DNS サービスを使用するため、他に DNS サービスが動作しているとインストール時にエラーが発生します。次のコマンド DNS サービスの動作を確認します。</p> <pre>sudo lsof -i :53</pre>
 <pre>[vm_admin@haa01 ~]\$ sudo vi /etc/systemd/resolved.conf</pre>	<p>左画像のように systemd-resolved のエントリが返される場合は次の手順で無効化します。</p> <pre>sudo vi /etc/systemd/resolved.conf</pre> <p>を実行し、下記行を追加します。</p> <p><code>DNSStubListener=no</code></p>
 <pre>[vm_admin@haa01 ~]\$ sudo systemctl restart systemd-resolved [vm_admin@haa01 ~]\$ sudo lsof -i :53 [vm_admin@haa01 ~]\$</pre>	<p><code>sudo systemctl restart systemd-resolved</code> を実行した後、再度 <code>sudo lsof -i :53</code> を実行し、systemd-resolved のエントリが返されないことを確認します。</p>
 <pre>[vm_admin@haa01 ~]\$ sudo kill &lt;dnsmasq-PID&gt; [vm_admin@haa01 ~]\$</pre>	<p>dnsmasq が実行されている場合には <code>sudo kill &lt;dnsmasq-PID&gt;</code> を実行した後、再度 <code>sudo lsof -i :53</code> を実行し、dnsmasq のエントリが返されないことを確認します。</p> <p>日本語ロケール環境ではインストール時にエラーが発生する既知問題があるため、<code>localectl</code> を実行し、System Locale が <code>LANG=en_US.UTF-8</code> になっていることを確認します。</p> <p><code>en_US.UTF-8</code> 以外の場合には次のコマンドを実行します。</p> <pre>export LANG="en_US.utf8"</pre>

```
[File Edit Setup Control Window KanjiCode Help]
[vm_admin@haa01 ~]$ sudo firewall-cmd --state
not running
[vm_admin@haa01 ~]$ sudo systemctl start firewalld
[vm_admin@haa01 ~]$
```

ファイアウォールが有効化されていることを確認します。

`sudo firewall-cmd --state`

“not running” が返される場合には次のコマンドでファイアウォールを有効化します。

`sudo systemctl start firewalld`

```
[File Edit Setup Control Window KanjiCode Help]
[vm_admin@haa01 ~]$ sudo swapoff -a
[vm_admin@haa01 ~]$ rm /tmp/haa
[vm_admin@haa01 ~]$ cd /tmp/haa
[vm_admin@haa01 haa]$ wget https://download.uipath.com/haa/get-haa.sh
--2023-05-24 02:14:37-- https://download.uipath.com/haa/get-haa.sh
Resolving download.uipath.com (download.uipath.com)... 104.19.252.9, 104.19.251.9, 2606:4700::6013:fc09, ...
Connecting to download.uipath.com (download.uipath.com)|104.19.252.9|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10316 (10K) [application/x-shellscript]
Saving to: 'eget-haa.sh'

get-haa.sh          100%[=====]  10.07K   --.-K/s   in 0s

2023-05-24 02:14:37 (51.5 MB/s) - eget-haa.sh saved [10316/10316]
[vm_admin@haa01 haa]$ chmod +x get-haa.sh
[vm_admin@haa01 haa]$
```

swap を無効化します。

`sudo swapoff -a`

`sudo sed -i.bak '/ swap / s/^(.*$/#1/g' /etc/fstab`

作業ディレクトリを作成します。

`mkdir /tmp/haa`

`cd /tmp/haa`

【インターネット接続が可能な環境】では HAA インストール用スクリプトをダウンロードし、実行権限を付与します。

`wget https://download.uipath.com/haa/get-haa.sh`

`chmod +x get-haa.sh`

次に [1台目の HAA インストール](#) を実行します。

【インターネットアクセスが不可の環境】では [オフライン環境での HAA インストール](#) の手順を実行します。

### 3.2. 1台目の HAA インストール

- 前節の続き、HAA ホストに SSH クライアントでアクセスした上で実行します。

1台目のHAAをインストールします。

**【DNS サーバー未使用時】** <admin-email> には管理者のメールアドレス、<haa-password> には HAA 管理コンソールにログインするためのパスワードを指定します。

```
sudo ./get-haa.sh -u <admin-email> -p  
<haa-password> -o Rhel8 --accept-  
license-agreement
```

<haa-password> に \$ などのシェル特殊文字を使用する場合には 'pa\$\$w0rd!' のようにシングルコーテーションで囲みます。

【DNS サーバー使用時】 -d オプション引数<haa-cluster-fqdn>にて HAA クラスタの FQDN を指定します。

```
sudo ./get-haa.sh -u <admin-email> -p  
<haa-password> -o Rhel8 -d <haa-cluster-  
fqdn> --accept-license-agreement
```

左画像のようにメッセージが表示された場合には  
正常にインストールされています。

途中エラーが発生する場合には、[デバッグ情報収集](#)をご参照ください。

**【HAA シングル構成の場合】**には、[HAA 管理コンソールでの操作](#)を参照の上、HAA 管理コンソールでの操作を行います。

### 3.3. 2台目以降の HAA インストール

- 【HAA 冗長構成の場合】には 2 台目以降においても [HAA インストール前準備](#) を実行し、以降、HAA ホストに SSH クライアントでアクセスした上で実行します。

```
[#] ./bin/stop_ctl.sh haagent EnvConfig http
[ha_admin@ha02 haagent]$ sudo -p 'get-haa-sh -u haadmin@example.com -p pa$$w0rd!' -a RHEL8 -j 10.0.0.8 --accept-license-agreement
[INFO] [2023-05-24T08:51:59+0000] Configuring High Availability Add-On
[INFO] [2023-05-24T08:51:59+0000] Downloading HA add-on package from https://cluster-02.4.2.249.15.193:3022/4.2.2/tarball/haam-3022.4.2.2/tarball/haam-3022.4.2.2.tar.gz
Resolving download.upipath.com... 104.19.252.9, 104.19.252.9, 2006:4700::6813:fb09, ...
Connecting to download.upipath.com [download.upipath.com][104.19.252.9]:443... connected.
HTTP/2 200 OK
Content-Type: application/x-tar
Content-Length: 135517193 [application/x-tar]
Saving to: /ehaa-2022.4.2.tar.gz [100%]
Saving: /ehaa-2022.4.2.tar.gz [100%] 128.86M 180MB/s in 0.7s

[ha-2022.4.2.tar.gz] 100%[=====] 128.86M 180MB/s in 0.7s

[2023-05-24 04:51:59 (100 MB/s) - /ehaa-2022.4.2.tar.gz f saved [135517193/135517193]

[INFO] [2023-05-24T08:51:59+0000] Decompressing archive
[INFO] [2023-05-24T08:52:00+0000] Installing High Availability Add-On
[INFO] [2023-05-24T08:55:54+0000] Waiting for the API to be ready...
[INFO] [2023-05-24T08:55:54+0000] Waiting for the API to be ready...
[INFO] [2023-05-24T08:55:54+0000] Joining node to Cluster... 10.0.8.8
HTTP/1.1 200 OK
server: envoy
date: Tue, 24 May 2023 04:55:37 GMT
content-type: text/html; charset=utf-8
content-length: 0
x-envoy-upstream-service-time: 435
x-content-type-options: noneif

[1/12] Waiting for the node to join the cluster...
[2/12] Waiting for the node to join the cluster...
[3/12] Waiting for the node to join the cluster...
[4/12] Waiting for the node to join the cluster...
[5/12] Waiting for the node to join the cluster...
[6/12] Waiting for the node to join the cluster...
[7/12] Waiting for the node to join the cluster...
[8/12] Waiting for the node to join the cluster...
[9/12] Waiting for the node to join the cluster...
[10/12] Waiting for the node to join the cluster...
[11/12] Waiting for the node to join the cluster...
[12/12] Waiting for the node to join the cluster...
[INFO] [2023-05-24T08:55:57+0000] "start_time": "2023-05-24T04:55:37Z", "state": "completed", "local_ip": "10.0.8.8", "node_id": "node-002", "architectures": "x86_64", "available_addresses": "[\"10.0.0.10\"], \"format\": \"ipv4\", \"if_name\": \"eth0\", \"private_ip\": \"true\", \"bigipids_storage_path\": \"var/opt/radisalbs/Flash\", \"cme_version_sha\": \"d8bc0e2492db065344ba09bbab7208c9b\""}]
```

2台目のHAAをインストールします。<admin-email>には管理者のメールアドレス、<haa-password>にはHAAの管理コンソールのログインするためのパスワードを、1台目インストール時と同じ物を指定します。

【DNS サーバー未使用時】 -j オプション引数

<haa01-ip-address> には 1 台目の HAA ホストの IP アドレスを指定します。

```
sudo ./get-haa.sh -u <admin-email> -p  
<haa-password> -o Rhel8 -j <haa01-ip-  
address> --accept-license-agreement
```

【DNS サーバー使用時】 -j オプション引数 <haa01-ip-address> には 1 台目の HAA ホストの IP アドレス、 -d オプション引数 <haa-cluster-fqdn> にて HAA クラスタの FQDN を指定します。

```
sudo ./get-haa.sh -u <admin-email> -p  
<haa-password> -o Rhel8 -j <haa01-ip-  
address> -d <haa-cluster-fqdn> --accept-  
license-agreement
```

2台目のインストール成功時には左画像のようにメッセージが表示されます。

3台目のHAAも2台目と同様の手順でインストールします。

### 3.4. オフライン環境での HAA インストール

- 【インターネットアクセスが不可の環境】では次の手順にて HAA インストールを実行します。

```
[vm_admin@haa02 ~]$ cd /mnt/cdrom/BaseOS/Packages
[vm_admin@haa02 Packages]$ sudo rpm -ivh cyrus-sasl-md5-2.1.27-6.el8_5.x86_64.rpm
sudo rpm -ivh cyrus-sasl-scram-2.1.27-6.el8_5.x86_64.rpm
sudo rpm -ivh cyrus-sasl-plain-2.1.27-6.el8_5.x86_64.rpm
Verifying... #####
Preparing... #####
Updating / installing...
 1:cyrus-sasl-md5-2.1.27-6.el8_5 #####
[vm_admin@haa02 Packages]$ sudo rpm -ivh cyrus-sasl-scram-2.1.27-6.el8_5.x86_64.rpm
Verifying... #####
Preparing... #####
Updating / installing...
 1:cyrus-sasl-scram-2.1.27-6.el8_5 #####
[vm_admin@haa02 Packages]$ sudo rpm -ivh cyrus-sasl-plain-2.1.27-6.el8_5.x86_64.rpm
Verifying... #####
Preparing... #####
package cyrus-sasl-plain-2.1.27-6.el8_5.x86_64 is already installed
[vm_admin@haa02 Packages]$ sudo rpm -ivh cyrus-sasl-2.1.27-6.el8_5.x86_64.rpm
Verifying... #####
Preparing... #####
Updating / installing...
 1:cyrus-sasl-2.1.27-6.el8_5 #####
[vm_admin@haa02 Packages]$ cd /mnt/cdrom/AppStream/Packages
[vm_admin@haa02 Packages]$ sudo rpm -ivh boost-program-options-1.66.0-10.el8.x86_64.rpm
Verifying... #####
Preparing... #####
Updating / installing...
 1:boost-program-options-1.66.0-10.el8.x86_64 #####
[vm_admin@haa02 Packages]$
```

下記の前提条件の RPM パッケージをあらかじめインストールします。

- cyrus-sasl-md5
- cyrus-sasl-scram
- cyrus-sasl-plain
- cyrus-sasl
- boost-program-options

これらを RHEL 8.6 のインストールメディア (rhel-8.6-x86\_64-dvd.iso) からインストールする手順について説明します。メディアをあらかじめ挿入します。

コマンドでメディアをマウントします。

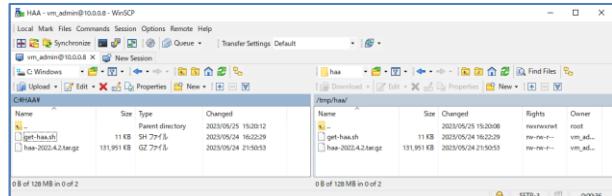
```
sudo mkdir /mnt/cdrom
sudo mount -t iso9660 /dev/cdrom
/mnt/cdrom
```

上記のパッケージを rpm コマンドでインストールします。パッケージファイル名は OS バージョンによって異なる点に注意してください。下記は RHEL 8.6 の例です。

```
cd /mnt/cdrom/BaseOS/Packages
sudo rpm -ivh cyrus-sasl-md5-2.1.27-6.el8_5.x86_64.rpm
sudo rpm -ivh cyrus-sasl-scram-2.1.27-6.el8_5.x86_64.rpm
sudo rpm -ivh cyrus-sasl-plain-2.1.27-6.el8_5.x86_64.rpm
sudo rpm -ivh cyrus-sasl-2.1.27-6.el8_5.x86_64.rpm
cd /mnt/cdrom/AppStream/Packages
sudo rpm -ivh boost-program-options-1.66.0-10.el8.x86_64.rpm
```

メディアをアンマウントします。

```
cd /tmp/haa
sudo umount /mnt/cdrom
```



インターネット接続されたマシンで次のファイルをダウンロードします。

<https://download.uipath.com/haa/get-haa.sh>

スクリプト内の HAA\_URL の値をコピーし、\$OS 部分を “Rhel8” に置き換えた URL からファイルをダウンロードします。

(例)

[https://download.uipath.com/haa/2022.4.2/\\$OS/haa-2022.4.2.tar.gz](https://download.uipath.com/haa/2022.4.2/$OS/haa-2022.4.2.tar.gz)

→

<https://download.uipath.com/haa/2022.4.2/Rhel8/haa-2022.4.2.tar.gz>

WinSCP などを使用して HAA ホストに接続し、get-haa.sh と haa-\*.tar.gz を /tmp/haa にコピーします。



HAA ホストにて SSH クライアントでアクセスした上で次のコマンドによりインストール用スクリプトに実行権限を付与します。

`chmod +x get-haa.sh`

以降の手順は root 権限にて実行します。

`sudo su -`

`mkdir /tmp/haa`

**【HAA 冗長構成の場合】** 2 台目、3 台目においても RPM パッケージのインストールから上記までの手順を実行します。

1台目のHAAをインストールします。オンラインでインストールするための--offlineオプションを使用します。

【DNS サーバー未使用時】<admin-email> には管理者のメールアドレス、<haa-password> には HAA 管理コンソールにログインするためのパスワードを指定します。

```
./get-haa.sh -u <admin-email> -p <haa-  
password> -o Rhel8 --offline --accept-  
license-agreement
```

<haa-password> に \$ などのシェル特殊文字を使用する場合には 'pa\$\$w0rd!' のようにシングルコーテーションで囲みます。

【DNS サーバー使用時】 -d オプション引数<haa-cluster-fqdn>にて HAA クラスタの FQDN を指定します。

```
./get-haa.sh -u <admin-email> -p <haa-  
password> -o Rhel8 --offline -d <haa-  
cluster-fqdn> --accept-license-agreement
```

```
[File Edit Setup Control Window KeyCode Help]
[root@ha02 haal]# ./get-haa.sh -u haas-admin@example.com -p "pa55w0rd!" -d haas-cluster.lab.test -j 10.0.0.9 -o offline --accept-license-agreement
[2023-05-26T08:08:56+0000]: Configuring High Availability Add-On
[2023-05-26T08:08:56+0000]: Decompressing archive
[2023-05-26T08:08:56+0000]: Installing High Availability Add-On
```

**【HAA 冗長構成の場合】** 2 台目の HAA をインストールします。<admin-email> には管理者のメールアドレス、<haa-password> には HAA の管理コンソールのログインするためのパスワードを、1 台目インストール時と同じ物を指定します。

【DNS サーバー未使用時】 -j オプション引数  
<haa01-ip-address> には 1 台目の HAA ホストの IP  
アドレスを指定します。

```
./get-haa.sh -u <admin-email> -p <haa-  
password> -o Rhel8 --offline -j <haa01-  
ip-address> --accept-license-agreement
```

【DNS サーバー使用時】 -j オプション引数 <haa01-ip-address> には 1 台目の HAA ホストの IP アドレス、 -d オプション引数<haa-cluster-fqdn>にて HAA クラスタの FQDN を指定します。

```
./get-haa.sh -u <admin-email> -p <haa-  
password> -o Rhel8 --offline -j <haa01-  
ip-address> -d <haa-cluster-fqdn> --  
accept-license-agreement
```

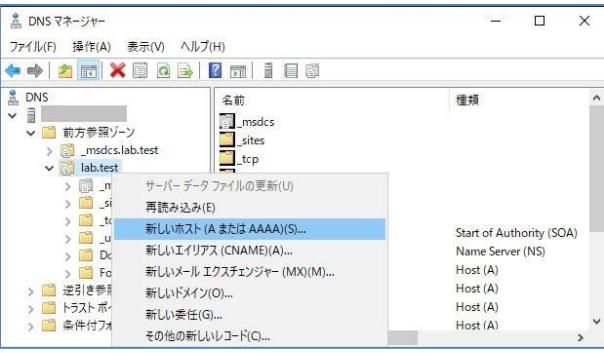
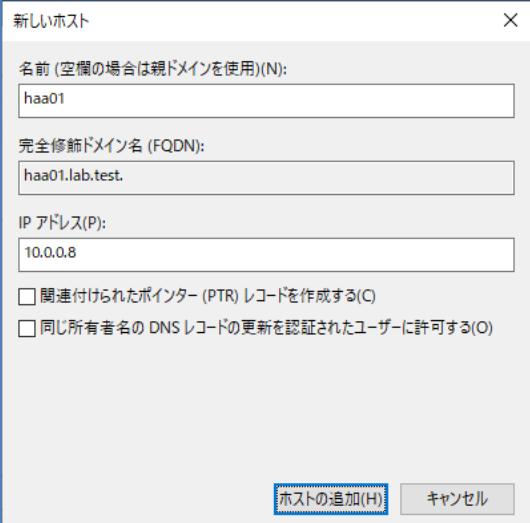
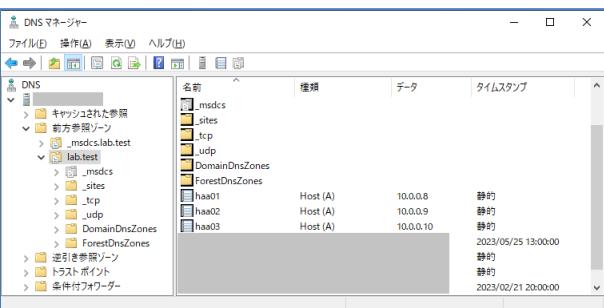
```
[In ] iH Setup Control [Window] EnvCode http
[http://127.0.0.1:30000:3242:00000] Activating DB Replication...
HTTP/1.1 200 OK
server: envoy
date: Fri, 26 May 2023 03:32:43 GMT
content-type: application/json
content-length: 1033
cluster-state-id: 16
x-envoy-upstream-service-time: 203
{"acl":[],"action_id": "5fb7183-4db6-b43-08aa-bf4b0ff","af_policy": "appendifsync-envoy-sec","authentication_admin_pass": "txZm3Wt0x01X0ycy1ax0qWaC7w9vJfhebtJpBsl","authentication_redis_pass": "pa5$4dv!l","authentication_ssl_pass": "", "authentication_ssl_uname": "", "authentication_ssl_client_certs": [], "authentication_ssl_crt_certs": [], "authorized_names": []}, "auto_upgrade": false, "background_op": [{"name": "updateDB", "progress": 100, "status": "done"}], "backup": false, "base64": false, "bigstore": false, "bigstore_ram_size": 0, "crdt": false, "crdt_causal_consistency": false, "crdt_config_version": "0", "crdt_ghost_replica_ids": "", "crdt_guid": "", "crdt_modules": [""], "crdt_repl_backlog_size": "auto", "crdt_replica_id": 0, "crdt_replicas": [], "crdt_sync": "disabled", "crdt_sync_dist": false, "crdt_syncer_auto_com_umatch": true, "crdt_syncer_max_retry": 10, "crdt_syncer_min_retry": 1, "crdt_syncer_timeout": 1000, "crdt_syncer_update": false, "crdt_sources": []}, "default_user": true, "dns_address_master": "", "dns_suffixes": [], "email_alerts": false, "endpoints": [{"addr": "[::1:0.4]", "addr_type": "external"}, {"dn_name": "redis-10800.has-cluster.lab.test", "oss_cluster_api_preferred_ip_type": "internal"}, {"port": 10800}], "proxy_policy": "single", "uid": "73-1"), "eviction_policy": "volatile-lru", "generate_text": "monitor", "false", "gossip": false, "group_id": 0, "hash_slots_policy": "10k", "implicit_shard_key": false, "import_failure_reason": "", "import_progress": "0.0", "import_status": "", "internal": false, "last_changed_time": "2023-05-26T00:24:29Z", "master_persistence": false, "max_af_file_size": "322125472900", "max_af_log_file_size": "3600", "max_connections": "memory_size/2147483648", "metrics_export_all": false, "mkeys": true, "node_id": 0, "node_name": "redis-10800", "node_type": "redis", "oss_cluster": false, "oss_cluster_ha": false, "oss_cluster": false, "oss_cluster_api_preferred_ip_type": "internal", "oss_sharding": false, "port": "10800", "proxy_policy": "single", "rack_aware": false, "redis_version": "6.0", "rep1_backlog_size": "auto", "replica_sources": [], "replica_sync": "disabled", "replica_sync_dist": false, "replica_type": "replicator", "role": "replicator", "routable_permissions": "hard-blocked,not-key-disse,not-blocked"}, {"replica_type": "replicator"}, {"shard_id": 0, "shard_name": "1", "shard_type": "slave"}, {"redis_cluster": 1, "replica_ip": "192.168.1.10", "shard_id": 0, "shard_name": "1", "shard_type": "slave"}, {"skip_import_analyze": "disabled", "slave_buffer": "auto", "slave_ha": true, "slave_ha_priority": 0, "snapshot_policy": "[]", "ssl": "false", "status": "active-change-pending", "sync": "disabled", "sync_sources": "[]", "syncer_mode": "centralized", "tls_mode": "disabled", "type": "redis", "uid": "3", "version": "6.0.13", "wait_command": true}
[Out ] iH Setup Control [Window] EnvCode http
[http://127.0.0.1:30000:3242:00000] Installation and Configuration of High Availability Add-On is finished.
[root@had01 ~]
```

2台目のインストール成功時には左画像のようにメッセージが表示されます。

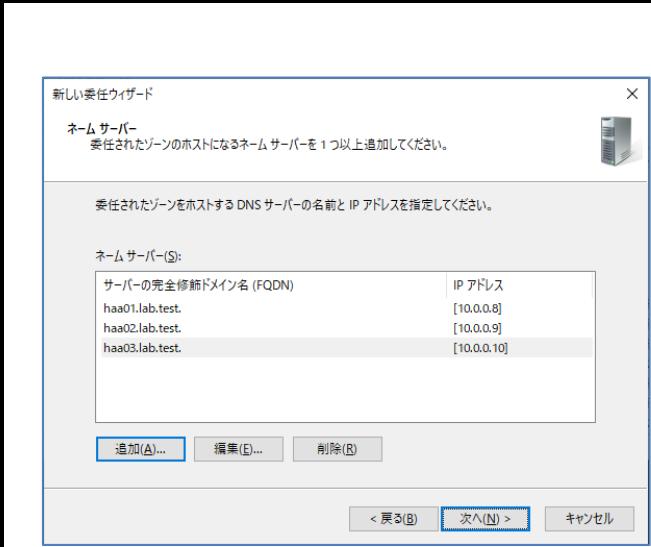
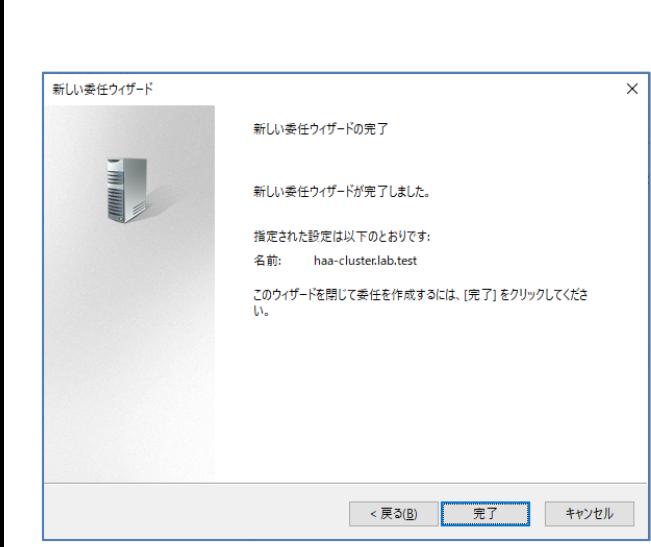
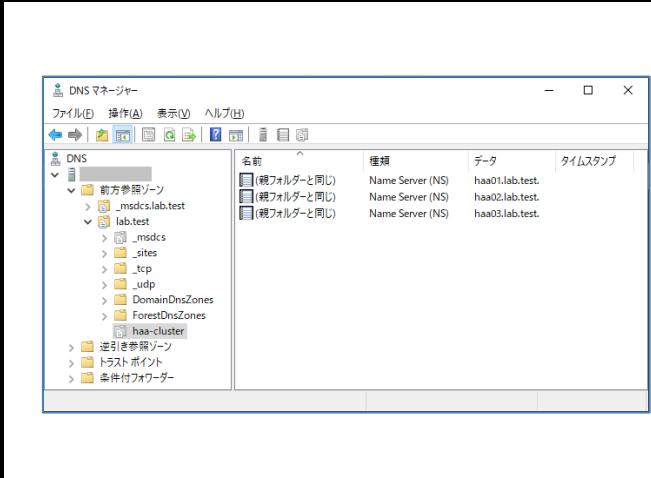
3台目のHAAも2台目と同様の手順でインストールします。

### 3.5. DNS サーバーの設定

- 【DNS サーバー使用時】HAA 冗長構成では DNS サーバーの NS レコードを使用して、HAA 接続のエンドポイントを個別の IP アドレスではなく、たとえば redis-10000.haa-cluster.lab.test のように 1 つの FQDN に集約することができます。このエンドポイントは現在マスターとなっている HAA ノードに自動的に名前解決されます。
- HAA 冗長化において DNS サーバーを使用する場合は、次の手順により NS レコードを作成します。ここでは Microsoft DNS サーバーを使用します。

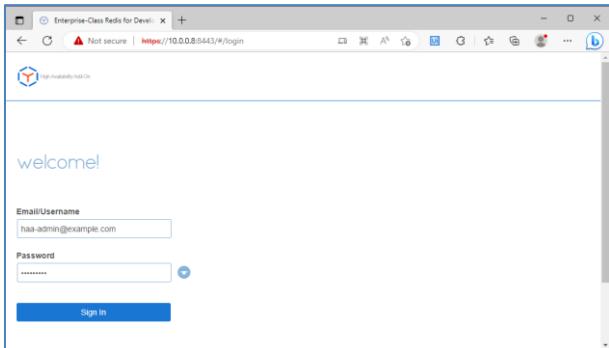
 <p>DNS マネージャーを開き、適切な前方参照ゾーンにて「新しいホスト」をクリックし、各 HAA ノードの A レコードを作成します。</p>	
 <p>ホスト名と IP アドレスを入力します。 3 つのノードに対して同じ手順を繰り返し、3 つの A レコードが作成されることを確認します。</p>	
	

	<p>次に DNS マネージャーにて「新しい委任」をクリックし、新しい委任ウィザードを開始します。</p>
	<p>「委任されたドメイン名」として、あらかじめ策定した HAA クラスタのホスト名を指定します。</p>
	<p>ネームサーバーを追加するために [追加] をクリックします。</p> <p>HAA ノードそれぞれについて、上記手順にて A レコードを作成した際に指定したホスト名 + ドメイン名 (例: haa01.lab.test) をサーバーの完全修飾ドメイン名に入力し [解決] をクリックします。検証済みが「OK」と表示されることを確認し、[OK] をクリックします。</p>

	<p>上記手順を 3 つのノードに対して繰り返し、3 つのネームサーバーが追加されたことを確認し、[次へ] をクリックします。</p>
	<p>委任ウィザードを完了します。</p>
	<p>委任が追加され 3 つの NS レコードが作成されていることを確認します。</p>

### 3.6. HAA 管理コンソールでの操作

- 以降、ブラウザーで操作します。

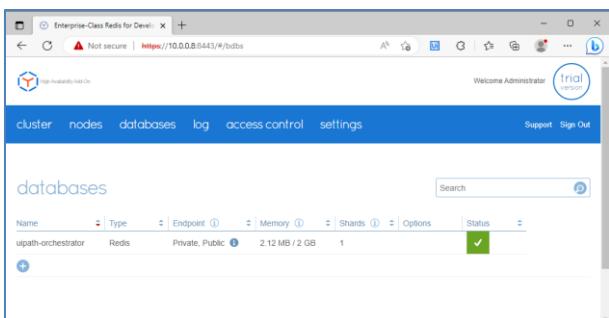


ブラウザーで `https://<haa-ip-address>:8443` にアクセスし、管理コンソールを開きます。

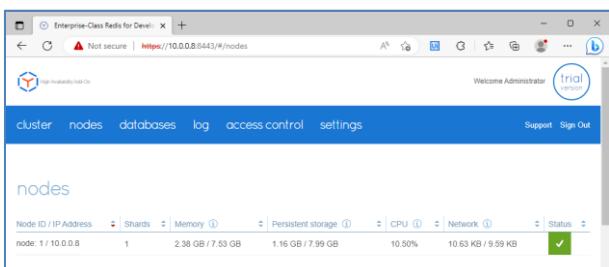
`<haa-ip-address>` は HAA ホストの IP アドレス (冗長構成の場合は 1 台目) を指定します。

インストール時に指定した`<admin-email>`と`<haa-password>`を指定してログインします。

**【DNS サーバー使用時】** FQDN を使用してアクセスすることができます。(例: `https://haa-cluster.lab.test:8443`)

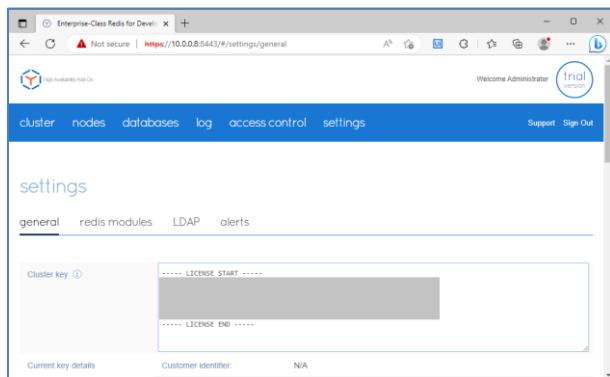


[databases] をクリックし、DB が正常に作成されていることを確認します。



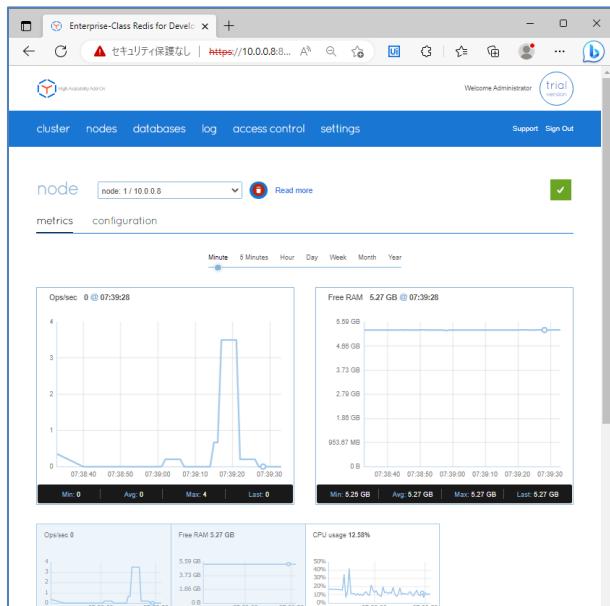
[nodes] をクリックし、ノードの Status がグリーンであることを確認します。

**【HAA 冗長構成の場合】** にはすべてのノードが表示されることを確認します。



HAA ライセンスをお持ちの方は次の手順でライセンスをインストールします。

[settings] > general > Cluster key に入手したライセンスコードを入力して Save します。



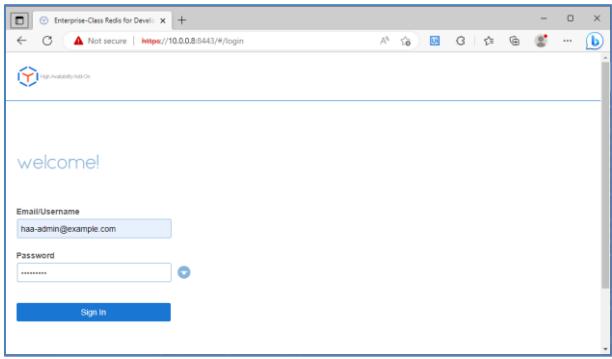
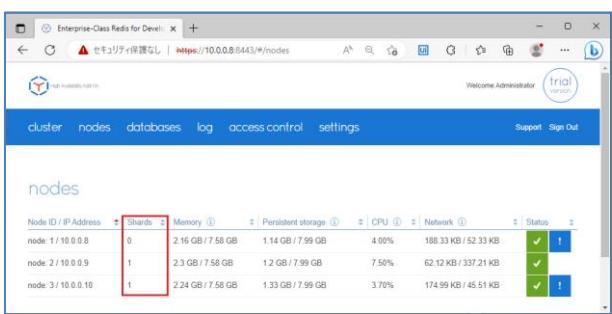
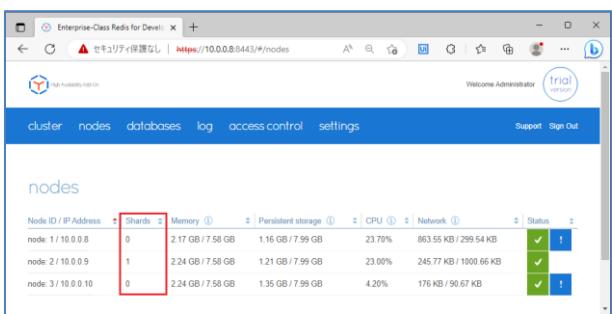
下記のページより、それぞれのメトリックスをリアルタイムで確認することができます。

- クラスタ状態: [cluster] > metrics
- ノード状態: [nodes] > (ノード選択) > metrics
- DB 状態: [databases] > (DB 選択) > metrics

- 【HAA シングル構成の場合】は、次に [Orchestrator の HAA 接続設定](#) を参照し、Orchestrator ホストにて HAA の接続設定を行います。

### 3.7. HAA 管理コンソールでの冗長構成確認

- 【HAA 冗長構成の場合】は正常に構成されているかを確認します。以降、ブラウザーで操作します。

	<p>ブラウザーで <code>https://&lt;haa01-ip-address&gt;:8443</code> にアクセスし、管理コンソールを開きます。  <code>&lt;haa01-ip-address&gt;</code> は HAA ホスト 1 台目の IP アドレスを指定します。  インストール時に指定した <code>&lt;admin-email&gt;</code> と <code>&lt;haa-password&gt;</code> を指定して、ログインします。</p>																												
<p><b>【レプリカあり】</b></p>  <table border="1"> <thead> <tr> <th>Node ID / IP Address</th> <th>Shards</th> <th>Memory</th> <th>Persistent storage</th> <th>CPU</th> <th>Network</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>node 1 / 10.0.0.8</td> <td>0</td> <td>2.16 GB / 7.58 GB</td> <td>1.14 GB / 7.99 GB</td> <td>4.00%</td> <td>188.33 KB / 52.33 KB</td> <td>!</td> </tr> <tr> <td>node 2 / 10.0.0.9</td> <td>1</td> <td>2.3 GB / 7.58 GB</td> <td>1.2 GB / 7.99 GB</td> <td>7.50%</td> <td>62.12 KB / 337.21 KB</td> <td>✓</td> </tr> <tr> <td>node 3 / 10.0.0.10</td> <td>1</td> <td>2.24 GB / 7.58 GB</td> <td>1.33 GB / 7.99 GB</td> <td>3.70%</td> <td>174.99 KB / 45.51 KB</td> <td>!</td> </tr> </tbody> </table>	Node ID / IP Address	Shards	Memory	Persistent storage	CPU	Network	Status	node 1 / 10.0.0.8	0	2.16 GB / 7.58 GB	1.14 GB / 7.99 GB	4.00%	188.33 KB / 52.33 KB	!	node 2 / 10.0.0.9	1	2.3 GB / 7.58 GB	1.2 GB / 7.99 GB	7.50%	62.12 KB / 337.21 KB	✓	node 3 / 10.0.0.10	1	2.24 GB / 7.58 GB	1.33 GB / 7.99 GB	3.70%	174.99 KB / 45.51 KB	!	<p>[nodes] をクリックし、3つのノードの Status がグリーンであることを確認します。  次に Shards フィールドを確認します。  左画像の 【レプリカあり】 のように node 1 では 1、node 2 または node 3 が 1 となっていれば DB が正常に冗長化されています。</p>
Node ID / IP Address	Shards	Memory	Persistent storage	CPU	Network	Status																							
node 1 / 10.0.0.8	0	2.16 GB / 7.58 GB	1.14 GB / 7.99 GB	4.00%	188.33 KB / 52.33 KB	!																							
node 2 / 10.0.0.9	1	2.3 GB / 7.58 GB	1.2 GB / 7.99 GB	7.50%	62.12 KB / 337.21 KB	✓																							
node 3 / 10.0.0.10	1	2.24 GB / 7.58 GB	1.33 GB / 7.99 GB	3.70%	174.99 KB / 45.51 KB	!																							
<p><b>【レプリカなし】</b></p>  <table border="1"> <thead> <tr> <th>Node ID / IP Address</th> <th>Shards</th> <th>Memory</th> <th>Persistent storage</th> <th>CPU</th> <th>Network</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>node 1 / 10.0.0.8</td> <td>0</td> <td>2.17 GB / 7.58 GB</td> <td>1.16 GB / 7.99 GB</td> <td>23.70%</td> <td>863.55 KB / 299.54 KB</td> <td>✓</td> </tr> <tr> <td>node 2 / 10.0.0.9</td> <td>1</td> <td>2.24 GB / 7.58 GB</td> <td>1.21 GB / 7.99 GB</td> <td>23.00%</td> <td>245.77 KB / 1000.66 KB</td> <td>✓</td> </tr> <tr> <td>node 3 / 10.0.0.10</td> <td>0</td> <td>2.24 GB / 7.58 GB</td> <td>1.35 GB / 7.99 GB</td> <td>4.20%</td> <td>176 KB / 90.67 KB</td> <td>✓</td> </tr> </tbody> </table>	Node ID / IP Address	Shards	Memory	Persistent storage	CPU	Network	Status	node 1 / 10.0.0.8	0	2.17 GB / 7.58 GB	1.16 GB / 7.99 GB	23.70%	863.55 KB / 299.54 KB	✓	node 2 / 10.0.0.9	1	2.24 GB / 7.58 GB	1.21 GB / 7.99 GB	23.00%	245.77 KB / 1000.66 KB	✓	node 3 / 10.0.0.10	0	2.24 GB / 7.58 GB	1.35 GB / 7.99 GB	4.20%	176 KB / 90.67 KB	✓	<p>左画像の 【レプリカなし】 のように node 1 では 1、node 2 も node 3 も 0 となっている場合には DB が冗長化されていません。次の手順を実行します。</p>
Node ID / IP Address	Shards	Memory	Persistent storage	CPU	Network	Status																							
node 1 / 10.0.0.8	0	2.17 GB / 7.58 GB	1.16 GB / 7.99 GB	23.70%	863.55 KB / 299.54 KB	✓																							
node 2 / 10.0.0.9	1	2.24 GB / 7.58 GB	1.21 GB / 7.99 GB	23.00%	245.77 KB / 1000.66 KB	✓																							
node 3 / 10.0.0.10	0	2.24 GB / 7.58 GB	1.35 GB / 7.99 GB	4.20%	176 KB / 90.67 KB	✓																							

The screenshot shows the 'databases' section of the Redis configuration interface. A single database named 'uipath-orchestrator' is listed. It is a Redis database with a memory limit of 26 MB / 2 GB. The status is green.

【レプリカなし】[databases] をクリックし、"uipath-orchestrator" という名前の DB をクリックします。

The screenshot shows the 'configuration' section of the Redis configuration interface. It includes settings for Memory (Memory limit: 2.00 GB), Replication (Disabled), Redis Modules (None), Persistence (None), and various other Redis-specific configurations. At the bottom, there is an 'Edit' button highlighted with a red box.

[configuration] をクリックし、画面下の Edit をクリックします。

The screenshot shows the 'Edit' configuration dialog for the 'uipath-orchestrator' database. The 'Replication' checkbox is checked and highlighted with a red box. Other configuration options like Data persistence, Redis password, and Data eviction policy are also visible.

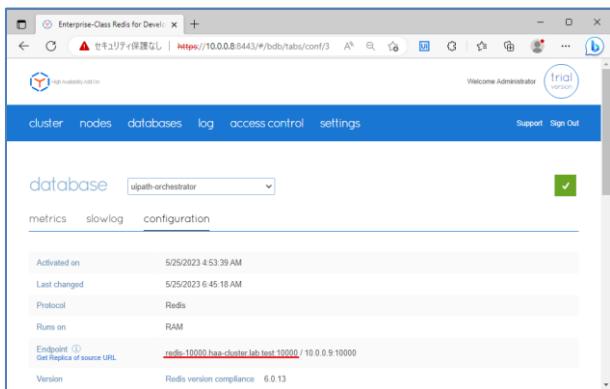
Replication のチェックをオンにし、Update をクリックします。

ハイパーバイザー環境上の HAA ホストに十分なメモリが割り当てられない場合に "Cannot allocate nodes for shards." というエラーが発生することがあります。その際には仮想マシンにあらかじめメモリを予約して割り当てるようハイパーバイザー上で設定変更します。

The screenshot shows the 'nodes' section of the Redis configuration interface. Three nodes are listed: node 1 (IP 10.0.0.8), node 2 (IP 10.0.0.10), and node 3 (IP 10.0.0.11). Node 1 has 1 shard, node 2 has 1 shard, and node 3 has 0 shards. The 'Shards' column is highlighted with a red box.

[nodes] をクリックし、Shards フィールドを再度確認します。

node 1 では 1、node 2 または node 3 が 1 となっていれば DB が正常に冗長化されています。



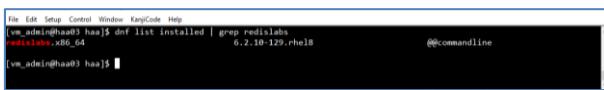
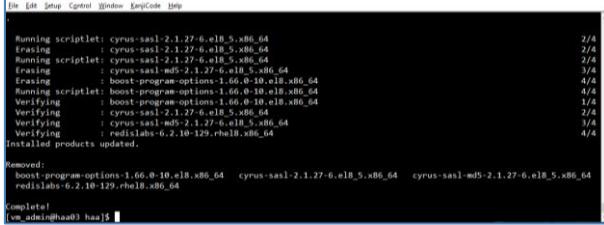
【DNS サーバー使用時】 [database] > uipath-orchestrator > configuration をクリックし、Endpoint をメモします。この値は Orchestrator から HAA 接続の設定を行う際に使用します。

なおこの Endpoint は現在マスターとなっている HAA ノードの IP アドレスに名前解決されます。マスターダウン時には次にマスターに昇格されたノードに名前解決されるため、自動的にフェールオーバーされます。

- 次に [Orchestrator の HAA 接続設定](#) を参照し、Orchestrator ホストにて HAA の接続設定を行います。

### 3.8. HAA アンインストール

- HAA インストールや設定が正常終了しなかった場合など、HAA をアンインストールするには、次の手順を実行します。以降、HAA ホストに SSH クライアントでアクセスした上で実行します。

	<p>HAA がインストール済みかは次のコマンドで確認します。インストール済みの場合には左画像のように一覧表示されます。</p> <pre>dnf list installed   grep redislabs</pre>
	<p>HAA アンインストールを実行するには次のコマンドを実行します。</p> <pre>sudo dnf remove redislabs -y</pre> <p>Complete! と表示されることを確認します。</p>

## 4. Orchestrator の HAA 接続設定

- Orchestrator で HAA に接続する設定を行うには、Orchestrator ホストに管理者でログインして次の手順を実行します。

### 【DNS サーバー未使用時】

```
C:\$Redis>redis-cli -h 10.0.0.8 -p 10000 -a pa$$wOrd! ping
PONG
C:\$Redis>.
```

### 【DNS サーバー使用時】

```
C:\$Redis>redis-cli -h redis-10000.haa-cluster.lab.test -p 10000 -a pa$$wOrd! ping
PONG
C:\$Redis>.
```

Windows 版 Redis の [ダウンロードサイト](#) から zip ファイルをダウンロードし、ローカルディレクトリに解凍します。redis-cli コマンドを使用して HAA への疎通確認を行います。

**【DNS サーバー未使用時】** <haa-ip-address>には HAA の IP アドレス(HAA 冗長構成の場合にはプライマリの IP アドレス)、<haa-password>には HAA のパスワードを指定します。HAA サービスはポート 10000 でリッスンするため -p でポート番号を指定します。

```
redis-cli -h <haa-ip-address> -p 10000 -a <haa-password> ping
```

正常接続された場合には "PONG" が返されます。

**【DNS サーバー使用時】** <haa-ip-address> の代わりにデータベースのエンドポイントを指定して "PONG" が返されることを確認します。

### 【HAA シングル構成の場合】

```
<add key="Auth.Password.DefaultExpirationDays" value="0" />
<add key="LoadBalancer.UseRedis" value="true" />
<add key="LoadBalancer.Enable" value="true" />
<add key="LoadBalancer.Redis.ConnectionString" value="10.0.0.8:10000,password=password" />
<add key="PingOne.SecretStores" value="<Orchestrator.CyberArk.dll><Orchestrator.AzureKeyVault.SecretStore.dll><Orchestrator.SemreStore.dll><Orchestrator.Orchestrator.SecretStore.BasicSecrets.dll><Orchestrator.Orchestrator.SecretStore.ReplayTrust.dll>" />
```

**【HAA シングル構成の場合】** Orchestrator の UiPath.Orchestrator.dll.config (C:\Program Files (x86)\UiPath\Orchestrator 配下) を下記のように設定します。

```
<add key="LoadBalancer.UseRedis" value="true" />
<add key="LoadBalancer.Redis.ConnectionString" value="<haa-ip-address>:10000,password=<haa-password>" />
```

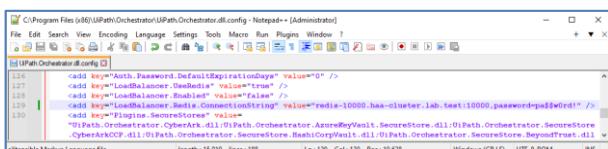
### 【HAA 冗長構成かつ DNS サーバー未使用時】

```
<add key="Auth.Password.DefaultExpirationDays" value="0" />
<add key="LoadBalancer.UseRedis" value="true" />
<add key="LoadBalancer.Enable" value="true" />
<add key="LoadBalancer.Redis.ConnectionString" value="10.0.0.8:10000,password=password" />
<add key="PingOne.SecretStores" value="<Orchestrator.CyberArk.dll><Orchestrator.AzureKeyVault.SecretStore.dll><Orchestrator.SemreStore.dll><Orchestrator.Orchestrator.SecretStore.BasicSecrets.dll><Orchestrator.Orchestrator.SecretStore.ReplayTrust.dll>" />
```

**【HAA 冗長構成かつ DNS サーバー未使用時】** には UiPath.Orchestrator.dll.config は次のように 3 ノードの IP アドレスをそれぞれ指定します。

```
<add key="LoadBalancer.UseRedis" value="true" />
<add key="LoadBalancer.Redis.ConnectionString" value="<haa01-ip-address>:10000,<haa02-ip-address>:10000,<haa03-ip-address>:10000,password=<haa-password>" />
```

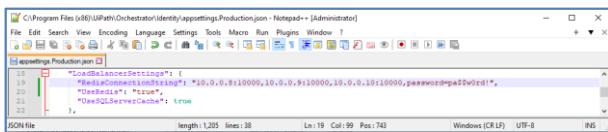
## 【HAA 冗長構成かつ DNS サーバー使用時】



## 【HAA シングル構成の場合】



## 【HAA 冗長構成かつ DNS サーバー未使用時】



## 【HAA 冗長構成かつ DNS サーバー使用時】



【HAA 冗長構成かつ DNS サーバー使用時】には `UiPath.Orchestrator.dll.config` は次のようにデータベースのエンドポイントを指定します。

```

<add key="LoadBalancer.UseRedis" value="true" />
<add key="LoadBalancer.Redis.ConnectionString" value=",password=<haa-password>" />
    
```

次に `appsettings.Production.json` (`C:\Program Files (x86)\UiPath\Orchestrator\Identity` 配下) を編集します。 `UiPath.Orchestrator.dll.config` で設定した接続文字列を下記箇所に指定します。

## 【HAA シングル構成の場合】

```

"LoadBalancerSettings": {
    "RedisConnectionString": "<haa-ip-address>:10000,
    password=<haa-password>",
    "UseRedis": "true"
}
    
```

【HAA 冗長構成かつ DNS サーバー未使用時】は 3 つの IP アドレスを指定します。

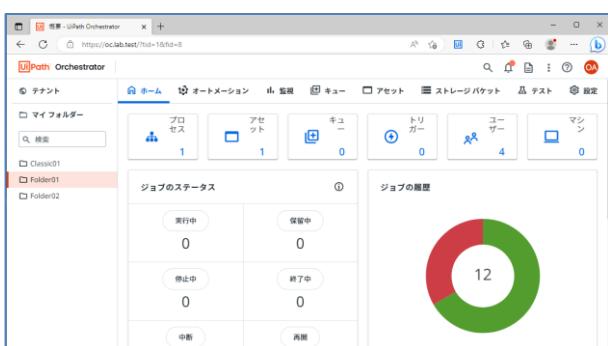
【HAA 冗長構成かつ DNS サーバー使用時】はデータベースのエンドポイントを指定します。

管理者権限コマンドプロンプトにて下記コマンドを実行し、IIS サービスを再起動します。

`iisreset`

Orchestrator のすべてのノードで上記設定を実行します。

ブラウザーで Orchestrator 管理コンソールを開き、正常にログインできることを確認します。



## 5. デバッグ情報収集

- ライセンスをご購入のお客様にはテクニカルサポートサービスを提供しております。上記以外のエラーが発生する場合または自己解決できない場合には、次の情報を取得し [UiPath カスタマーサポート](#)までご連絡ください。
- インストール時にエラーが発生する場合は、インストールを verbose モードで実行します。

```
sudo ./get-haa.sh -u <admin-email> -p <haa-password> -o Rhel8 --accept-license-agreement --verbose
```

- 出力されるメッセージをテキストファイルにコピーしたものをお送りください。(以下例)

```
[vm_admin@haa01 ~]$ sudo ./get-haa.sh -u haa-admin@example.com -p 'pa$$w0rd!' -d haa-cluster.lab.test --accept-license-agreement --verbose

[INFO] [2023-05-25T07:20:13+0000]: Configuring High Availability Add-On

[WARN] [2023-05-25T07:20:13+0000]: Installation type is online and no OS provided. Trying to determine the OS...

[INFO] [2023-05-25T07:20:13+0000]: Downloading High Availability Add-On on Rhel8

--2023-05-25 07:20:13-- https://download.uipath.com/haa/2022.4.2/Rhel8/haa-2022.4.2.tar.gz

Resolving download.uipath.com (download.uipath.com)... 104.19.251.9, 104.19.252.9, 2606:4700::6813:fc09,
...
Connecting to download.uipath.com (download.uipath.com)|104.19.251.9|:443... connected.

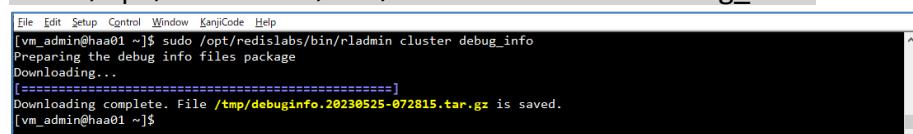
HTTP request sent, awaiting response... 200 OK

～中略～

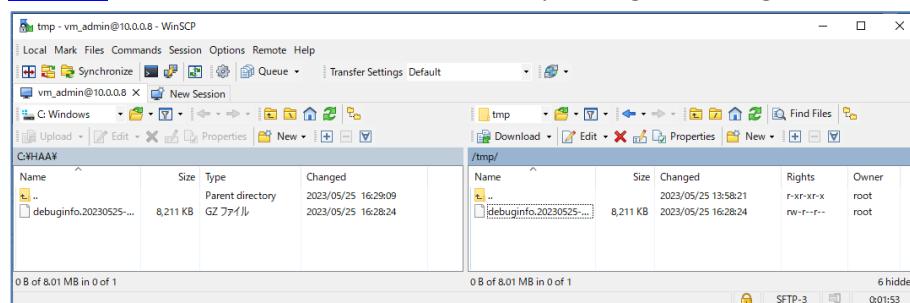
[INFO] [2023-05-25T07:22:09+0000]: Installation and Configuration of High Availability Add-On is finished.
```

- HAA 運用時にエラーなどが発生する場合には、SSH クライアントで HAA ホストにアクセスし、次のコマンドでデバッグ情報を /tmp に生成します。

```
sudo /opt/redislabs/bin/radmin cluster debug_info
```



- [WinSCP](#)などを使用して、生成された /tmp/debuginfo.\*.tar.gz を HAA ホストから転送します。



## 6. 技術支援のご案内

- UiPath 社では Orchestrator および周辺のテクノロジーに関する技術支援の有償コンサルティングサービスを提供しております。下記のような課題に対して技術支援が必要なお客様は弊社担当営業までご相談ください。
  - Orchestrator 設計・構築・運用
    - ✧ シングル構成または HAA を含む冗長構成での導入支援
    - ✧ オンプレミスまたはパブリッククラウド環境への導入支援
    - ✧ インストール後のフォルダー・ロール設定など運用ルール策定支援
  - Orchestrator / Studio / Robot のバージョンアップ
    - ✧ ベストプラクティスに基づくバージョンアップ作業手順の策定支援
  - UiPath Insights または Elasticsearch / Kibana 導入・活用
    - ✧ ダッシュボード作成によるログ可視化の活用支援

以上