Cisco WAE のインストール

ここでは、次の内容について説明します。

スーパバイザのインストールと設定

WAE をインストールする前に、スーパーバイザをインストールして設定します。

（注） 次の設定手順は、スーパーバイザが yum を使用してインストールされている場合にのみ機能します。他の方法を使用してスーパーバイザをインストールする場合は、**supervisorctl** を非ルートユーザーとして実行するように設定する必要があります。

ステップ 1 スーパーバイザをインストールして確認します。

```
sudo yum install -y epel-release
sudo yum install -y supervisor
supervisord -version
4.2.1
```

ステップ 2 WAE を実行している OS ユーザーの書き込み権限を持つディレクトリを作成します。

```
sudo mkdir -p /opt/supervisor/run
sudo mkdir -p /opt/supervisor/log
sudo chown -R [USER-NAME]:[GROUP-NAME] /opt/supervisor
```

ステップ 3 スーパーバイザ設定を、ルートユーザーとして実行しないように更新します。

pid ファイルを `/opt/supervisor/run/supervisor.pid` に指定し、ユーザーを WAE を実行している OS ユーザーとして指定します。

ルートとして `/etc/supervisord.conf` を開いて編集します。

- `[unix_http_server]` セクションで次を実行します。
  - `;file=/var/run/supervisor/supervisor.sock` を `file=/opt/supervisor/run/supervisor.sock` に変更します
  - `;chown=nobody:nogroup` を `chown=[USER-NAME]:[GROUP-NAME]` に変更します
- `[supervisord]` セクションで次を実行します。
  - `;logfile=/var/log/supervisor/supervisord.log` を `logfile=/opt/supervisor/log/supervisord.log` に変更します。
  - `;pidfile=/var/run/supervisord.pid` を `pidfile=/opt/supervisor/run/supervisord.pid` に変更します
  - `;minfds=1024` を `minfds=1000000` に変更します
  - `;minprocs=200` を `minprocs=257805` に変更します

  （注） `[supervisord]` セクションの下ではユーザーを設定しないでください。
- `[supervisorctl]` セクションで次を実行します。
  - `;serverurl=unix:///var/run/supervisor/supervisor.sock` を `serverurl=unix:///opt/supervisor/run/supervisor.sock` に変更します

ステップ 4 スーパーバイザを開始します。

```
sudo systemctl start supervisord
sudo supervisorctl status all
```

ステップ 5 システムの起動時にスーパーバイザが起動できるようにします。

```
sudo systemctl enable supervisord
sudo systemctl status supervisord
```

WAE イメージの確認

ステップ 1 Cisco ダウンロード ソフトウェア サイトから、Cisco WAE 7.6.0 ソフトウェアパッケージをダウンロードします。

ステップ 2 証明書とデジタル署名は両方ともダウンロードされたファイル（`wae-linux-v7.6.0.signed.bin`）に組み込まれています。

ステップ 3 自己解凍型の署名付きバイナリを実行します。実行するとリリースバイナリが抽出され、署名ファイルを使用して検証されます。

署名済みイメージの検証

```
[admin@wae-vm-21 workspace.signed]$ ./wae-linux-v7.6.0.signed.bin
Unpacking...
Verifying signature...
Downloading CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...
Successfully downloaded and verified crcam2.cer.
Downloading SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...
Successfully downloaded and verified innerspace.cer.
Successfully verified root, subca and end-entity certificate chain.
Successfully fetched a public key from WAE-CCO_RELEASE.cer.
Successfully verified the signature of wae-linux-v7.6.0.bin using WAE-CCO_RELEASE.cer
```

ステップ 4 生成された `wae-linux-v7.6.0.bin` は、WAE の Linux インストーラです。

Cisco WAE のインストール

始める前に

（注） 以前の WAE 7.x リリースから WAE 7.6.0 にアップグレードする場合は、Cisco WAE 7.x からのアップグレード （9 ページ） を参照してください。

- まだ存在していない場合は、（グループに割り当てられる）UNIXユーザーを作成します。インストールを実行するには、この UNIX ユーザーである必要があります。
- Java 11 および Python 3.6.x がシステムにインストールされていることを確認してください。`JAVA_HOME` 環境変数は jdk-11.0 を指し、`/usr/bin/python3` はインストールされた Python を指している必要があります。
- スーパーバイザがインストールおよび設定されていることを確認します。スーパバイザのインストールと設定 （1 ページ） を参照してください。
- デジタル署名された Cisco WAE 7.6.0 イメージをダウンロードして確認します。WAE イメージの確認 （2 ページ） を参照してください。
- WAE で BW-OPT アプリケーションが機能するように、`requests.auth` python パッケージがインストールされていることを確認してください。

ステップ 1 実行中の場合は WAE を停止します。

ステップ 2 次のコマンドを使用して、インストールファイルのアクセス許可を変更します。

Cisco WAE のインストール

```
chmod +x wae-linux-v7.6.0.bin
```

ステップ 3 ターゲットディレクトリを指定してインストーラを実行します。

```
./wae-linux-v7.6.0.bin <wae-dir>
```

ステップ 4 送信元 `waerc` のインストールディレクトリに移動します。環境を設定し、パスを指定してランタイムディレクトリを作成します。

```
cd  <wae-dir>
source waerc
wae-setup --dest <target-runtime-dir>
```

ステップ 5 Cisco WAE 管理者パスワードの設定を求められます。

```
WAE admin password:
Confirm password:
```

ステップ 6 WAE をインストールして設定した後（wae-setup の実行後など）、`/etc/supervisord.d/` 内から `wae.ini` ファイルへのソフトリンクを作成し、WAE 設定をスーパーバイザに追加します。

```
sudo ln -sf <target-runtime-dir>/wae.ini /etc/supervisord.d/
```

（注）
- この手順は、スーパーバイザがインストールおよび設定された後にのみ実行してください。
- `JAVA_HOME/JRE_HOME` を設定する必要がある external-executable-nimo ベースのネットワークを使用する場合は、`target-runtime-dir/wae.ini` ファイル内のセクション `[program:waectl]` を編集し、環境内に `JAVA_HOME="valid_jdk_path"` を含めます。

  たとえば、`[program:waectl]` の下で編集し、次を追加します。

  ```
  JAVA_HOME:environment=HOME="/home/wae", NCS_JAVA_VM_OPTIONS="-Xmx32G -Xms16G
  -XX:+UseG1GC -XX:+HeapDumpOnOutOfMemoryError
  -XX:HeapDumpPath=/home/wae/test/run/logs/ -Djava.io.tmpdir=/home/wae/test/run/work/",
   TMPDIR="/home/wae/test/run/work/", JAVA_HOME="/usr/"
  ```

- 新しい wae.ini の変更を有効にするには、**supervisorctl update** を実行します。

ステップ 7 スーパバイザ設定を更新します。

```
sudo supervisorctl update
```

ステップ 8 WAE プロセスの開始

```
sudo supervisorctl start wae:*
wae:zookeeper: started
wae:waectl: started
wae:kafka: started
wae:wae-monitor: started
```

（注）
- `wae:waectl` は WAE プログラムです。
- `wae:kafka` および `wae:zookeeper` は、トラフィックの収集と内部メッセージに必要です。
- `wae:wae-monitor` はモニタリングサービスです。
- `wae:logrotate` はログローテーション用です。

ステップ 9 WAE プロセスのステータスの確認

```
sudo supervisorctl status
wae:kafka RUNNING pid 1540, uptime 28 days, 14:03:40
wae:logrotate RUNNING pid 1178, uptime 28 days, 15:10:11
wae:wae-monitor RUNNING pid 11520, uptime 0:00:12
wae:waectl RUNNING pid 1177, uptime 28 days, 15:10:11
wae:zookeeper RUNNING pid 1736, uptime 28 days, 14:03:39
```

（注） すべての WAE プロセスを停止するには、次のコマンドを使用します。

```
sudo supervisorctl stop wae:*
```

ステップ 10 WAE 7.x.x リリースから WAE 7.6.0 リリースへ設定を移行するには、Cisco ダウンロード ソフトウェア サイトから、Cisco WAE アップグレードスクリプトを使用します。

（注） サーバー/VM を再起動すると、すべての WAE サービスは自動的に再起動されず、停止状態になります。WAE サービスは手順 8 で説明したコマンドを使用して開始できます。

Multi WAE のインストール

始める前に

- python3 ベースの Ansible バージョン 2.10.7 以降をインストールします。次のコマンドを使用します。

  ```
  sudo yum install ansible
  ```

- すべてのリモートホストに Java 11 をインストールします。
- すべてのリモートホストとプレイブックが実行されているホストに Python3 をインストールします。

（注）
- RHEL 8.4 では、**waerc** が提供されていないターミナルからプレイブックを実行します。
- スプリット数が増えるたびに、スケールプライマリで WAE を再起動します。

- マルチ WAE に参加しているサーバー間でパスワードなしの ssh を有効にします（セルフ ssh を含む）。

ステップ 1 ansible.cfg をエクスポートします。カスタムの ansible.cfg ファイルは、**playbooks/ansible.cfg** で提供されます。使用するコマンドは、次のとおりです。

```
export ANSIBLE_CONFIG=<path-to-the-ansible-config-file>
```

ステップ 2 プレイブックを実行する予定のマシンで、自分自身に SSH を実行し、**playbooks/known_hosts** ファイルの self にエントリを追加します。Multi WAE のインストールにより、異なるマシン間で単一の **username** と **wae_dir** のみがサポートされます。**-u** フラグを渡すと、**ansible-playbook** コマンドを呼び出すときに、CLI から **ansible_ssh_user** を渡すこともできます。

```
ansible-playbook wae_install.yml -u <username> --ask-pass
```

ステップ 3 **Playbooks/visudo** ファイルの最後に次の行を追加して、パスワードなしで sudo コマンドを実行できることを確認します。

```
<username> ALL=(ALL) NOPASSWD:ALL
```

ステップ 4 **Playbooks/hosts** ファイルを変更して、マシンの IP アドレスを含めます。**hosts** ファイルには、**[remote]**、**[primary]**、**[secondary]** の 3 つのグループがあります。

```
[remote]
'element-1' ansible_ssh_user='TARGET_SSH_USER'
'element-2' ansible_ssh_user='TARGET_SSH_USER'
'element-3' ansible_ssh_user='TARGET_SSH_USER'

[primary]
'element-1' ansible_ssh_user='TARGET_SSH_USER'

[secondary]
'element-2' ansible_ssh_user='TARGET_SSH_USER'

where,

[remote]    - indicate the set of hosts in which the playbooks are to be run
[primary]   - is the host which should be set as primary when configuring HA. Must be one of the
host present in [remote] group.
[secondary] - is the host which should be set as secondary when configuring HA. Must be one of the
 host present in [remote] group.
```

（注）
- **[remote]** グループは、すべてのプレイブックの実行に必須です。
- **[primary]** および **[secondary]** グループは、**ha_config** プレイブックの実行にのみ必要です。

ステップ 5 プレイブックに必要な入力パラメータを **group_vars/all** ファイルに設定します。このファイルは **playbooks/group_vars/all**（次の表を参照）にあり、プレイブックを実行します。次の表に、使用可能なプレイブックの詳細を示します。

表 *1* : *Ansible* プレイブックの詳細

プレイブック	説明	パラメータ	使用方法
wae_install.yml	**wae_install.yml** プレイブック は、WAE バイナリをコピーし、スーパーバイザを使用してサーバーを稼働させるために必要な関連チェックとタスクを実行することにより、リモートマシンに WAE をインストールします。	• **WAE_USER_NAME**：すべてのリモートマシンに既存の WAE ユーザー（sudo 対応）。 • **WAE_BIN_PATH**：**ansible-playbook** が実行されているマシンにある WAE バイナリへの絶対パス。 • **WAE_DIR**：**wae-install** および **wae-run** ディレクトリを保持する WAE ディレクトリの絶対パス。 • **DELETE_SIGNED**：インストールの完了後に署名された WAE イメージをクリーンアップする必要があるかどうか示すために使用されるフラグ。 デフォルト値は **False** です。	`ansible-playbook wae_install.yml -i <path_to_inventory_file> --ask-pass`
kafka_config.yml	**kafka_config.yml** プレイブックは、内部および外部リスナーに適切な設定を設定することにより、リモートマシンに kafka を展開します。	• **WAE_DIR**：**wae-install** および **wae-run** ディレクトリを保持する WAE ディレクトリの絶対パス。	`ansible-playbook kafka_config.yml -i <path_to_inventory_file> --ask-pass`

プレイブック	説明	パラメータ	使用方法
ha_config.yml	**ha_config.yml** プレイブックは、指定の WAE の実行中に 2 つのノード間で HA を展開します。	• **WAE_USER_NAME**：すべてのリモートマシンに既存の WAE ユーザー（sudo 対応）。 • **WAE_DIR**：**wae-install** および **wae-run** ディレクトリを保持する WAE ディレクトリの絶対パス。 • **WAE_HA_XML_TEMPLATE**：2 つのノードの CDB にロードされる WAE HA 設定を含む XML テンプレート。	`ansible-playbook ha_config.yml -i <path_to_inventory_file> --ask-pass`

プレイブック	説明	パラメータ	使用方法
load_config.yml	**load_config.yml** プレイブックは、リモート WAE サーバーに WAE 設定をロードすることを目的としています。	• **WAE_DIR**：**wae-install** および **wae-run** ディレクトリを保持する WAE ディレクトリの絶対パス。 • **WAE_CFGS_SRC_DIR**：ansible-playbook が実行されているマシン上に設定が存在するディレクトリの絶対パス。 • **WAE_CFGS**：設定ファイルの名前のリスト。ファイルは **WAE_CFGS_SRC_DIR** に存在する必要があります。 • **WAE_TMP_CFGS_DEST_DIR**：設定ファイルがコピーされるリモートマシンのディレクトリへの絶対パス。 存在しない場合は、ディレクトリが作成されます。デフォルト値は **/tmp/wae_cfgs** です。	`ansible-playbook load_config.yml -i <path_to_inventory_file> --ask-pass`

Cisco WAE 7.x からのアップグレード

始める前に

- Cisco ダウンロード ソフトウェア サイトから、Cisco WAE アップグレードスクリプトをダウンロードします。

- デジタル署名された Cisco WAE 7.6.0 イメージをダウンロードして確認します。WAE イメージの確認 （2 ページ） を参照してください。
- Java 11 および Python 3.6.x がシステムにインストールされていることを確認してください。`JAVA_HOME` 環境変数は jdk-11.0 を指し、`/usr/bin/python3` はインストールされた Python を指している必要があります。
- 次のコマンドを使用して、`pexpect` をインストールします。

  ```
  sudo pip3 install pexpect
  ```

- スーパーバイザがインストールおよび設定されていることを確認します。スーパバイザのインストールと設定 （1 ページ） を参照してください。
- アップグレードを実行する前に HA を無効にします。アップグレードスクリプトは、以前の WAE インストールに存在する特定の機能パックに関連する設定を処理しません。次の操作を実行できます。
  - アップグレードを実行する前に、機能パックに関連する設定を削除します。または、
  - WAE を手動でインストールし（Cisco WAE のインストール （3 ページ） を参照）、新しい WAE インストールに機能パックをインストールしてから、設定をインポートします（Cisco WAE 7.x からの設定の移行 （12 ページ） を参照）。

ステップ 1 7.x がインストールされたマシンにログインします。

ステップ 2 `wae_upgrade` スクリプトを実行します。

（注） `--wae-bin` オプションとして渡されるインストールファイルは、デジタル署名された Cisco WAE 7.6.0 イメージの検証後に取得されたイメージです。

```
# ./wae_upgrade --upgrade --old-install-dir <WAE_7.x_INSTALL_DIR> --old-run-dir <WAE_7.x_RUN_DIR>
 --new-install-dir <WAE_7.6.0_INSTALL_DIR> --new-run-dir <WAE_7.6.0_RUN_DIR>  --cfg-dir
<dir_to_save_config>  --wae-bin <WAE_7.6.0_INSTALLATION_FILE>
where
--old-install-dir    indicates the directory where 7.x WAE is installed
--old- run-dir       indicates the directory where the run time for 7.x WAE resides
--new-install-dir    indicates the directory where 7.6.0 WAE must be installed
--new-run-dir        indicates the directory where the run time for 7.6.0 WAE will reside
--cfg-dir            indicates the folder where the config is to be saved. This config will be
changed to match 7.6.0 and pushed to 7.6.0
--wae-bin            indicates the path to WAE 7.6.0 installation file.
```

Cisco WAE 7.x から Multi WAE へのアップグレード

ステップ 1 WAE インストールを Cisco WAE 7.6 にアップグレードします。Cisco WAE 7.x からのアップグレード （9 ページ） を参照してください。

ステップ 2 ansible playbook **load_config.yml** を使用してエージェントと NIMO を設定し、アップグレードされた WAE インスタンスで Multi WAE を手動で設定します。Multi WAE のインストール （5 ページ）を参照してください

ステップ 3 エクスポートオプションを指定してアップグレードスクリプトを実行し、更新された WAE インスタンスから設定を収集します。Cisco WAE 7.x からの設定の移行 （12 ページ） を参照してください。

ステップ 4 **wae_install** および **load_config** プレイブックを使用して、他の WAE インスタンスに WAE をインストールし、設定します。

Cisco WAE ライセンスのインストール

Cisco WAE のすべての機能を使用するには、ライセンスが必要です。ライセンスの取得について質問がある場合は、シスコのサポート担当者またはシステム管理者にお問い合わせください。

Cisco WAE は、Cisco Smart Licensing と従来のライセンスをサポートしています。従来のライセンスから Cisco Smart Licensing への切り替えを希望する場合は、Cisco WAE アカウント担当者にお問い合わせください。2 種類のライセンスの違いについては、Cisco.com で紹介している Cisco Smart Licensing の概要を参照してください。

Cisco Smart Licensing の詳細については、『*Cisco WAE* ユーザーガイド』の「Smart Licensing」の章を参照してください。

従来のライセンスのインストール

従来のライセンスをインストールするには、次の手順を実行します。

ステップ 1 `license_install` ツールを実行し、ライセンスファイルの名前（`.lic` 拡張子付き）を渡します。デフォルトでは、ツールによって、新しいライセンスで付与されたすべての機能と既存のライセンスの機能がマージされます。

```
license_install -file <path>/<license_name>.lic
```

ステップ 2 プロンプトが表示されたら、ライセンスをインストールするディレクトリに関連付けられた数字を入力します。

（注）
- オプション 2（<wae-dir>/etc）が選択されている場合、新しいビルドのインストール時にライセンスを再インストールする必要があります。
- オプション 1（/.cariden/etc）が選択されている場合、ライセンスの有効期限が切れていない限り、ライセンスを再インストールする必要はありません。
- ライセンスがインストールされたら、`license_check` コマンドを実行して、インストールされたライセンスを確認できます。

ステップ 3 WAE を停止してから起動し、インストールされているライセンスを取得します。

スマートライセンスのインストール

スマートライセンスをインストールするには、次の手順を実行します。

ステップ 1 スマートライセンスを設定するには、ユーザーガイドの「スマートライセンス」のセクションを参照してください。

ステップ 2 WAE を停止してから起動し、インストールされているライセンスを取得します。

Cisco WAE の開始と停止

Cisco WAE ランタイムディレクトリから関連する Cisco WAE CLI コマンドを入力して、Cisco WAE サービスを開始または停止します。

- WAE の開始

```
sudo supervisorctl start wae:*
wae:zookeeper: started
wae:waectl: started
wae:kafka: started
wae:wae-monitor: started
```

- WAE の停止

```
sudo supervisorctl stop wae:*
```

Cisco WAE 7.x からの設定の移行

Cisco WAE アップグレード スクリプト ユーティリティを使用して、WAE 7.x から設定を移行できます。

始める前に

- WAE 7.x から WAE 7.6.0 パッケージへ設定を移行するための Cisco WAE アップグレード スクリプトを、Cisco ダウンロード ソフトウェア サイトからダウンロードします。
- 設定の移行に進む前に、WAE 7.6.0 をインストールして WAE プロセスを開始します。Cisco WAE のインストール （3 ページ）を参照してください
- 次のコマンドを使用して、`pexpect` をインストールします。

```
sudo pip3 install pexpect
```

- アップグレードを実行する前に HA を無効にします。

- アップグレードスクリプトは、以前の WAE インストールに存在する特定の機能パックに関連する設定を処理しません。次の操作を実行できます。
  - エクスポートする前に、機能パックに関連する設定を削除します。または、
  - 設定をインポートする前に、新しい WAE インストールに機能パックをインストールします。

ステップ 1 7.x 設定のバックアップを取るには、7.x がインストールされているマシンにログインし、`--export` オプションを指定して `wae_upgrade` スクリプトを実行します。

```
# ./wae_upgrade --export --install-dir <WAE_7.x_INSTALL_DIR> --run-dir <WAE_7.x_RUN_DIR> --cfg-dir
 <dir_to_save_exported_config>
Where:
      --install-dir   indicates the directory where 7.x WAE is installed
      --run-dir       indicates the directory where the run time for 7.x WAE resides
      --cfg-dir       indicates the folder where backup of 7.x configuration must reside
```

ステップ 2 7.x 設定を7.6.0 に復元するには、7.6.0 がインストールされているマシンにログインし、`--import` オプションを指定して `wae_upgrade` スクリプトを実行します。

```
# ./wae_upgrade --import  --install-dir <WAE_7.6.0_INSTALL_DIR> --run-dir <WAE_7.6.0_RUN_DIR>
--cfg-dir <dir_to_import_saved_config>

Where:
      --install-dir   indicates the directory where 7.6.0 WAE is installed
      --run-dir       indicates the directory where the run time for 7.6.0 WAE resides
      --cfg-dir       indicates the folder where backup of 7.x configuration resides
```

パッケージまたはテンプレートの更新

<wae_run_time_directory>/packagesディレクトリに、パッケージまたはテンプレートが更新または追加された場合、Cisco 式の WAE CLI を使用してパッケージのリロードを要求します。

```
$ packages reload
```

たとえば、`wae.conf` ファイルを編集するときにパッケージのリロードを実行します。

Cisco WAE のインストールのトラブルシューティング

Cisco WAE のステータスを確認するには、`sudo supervisorctl status` と入力します。

Cisco WAE には YANG ランタイムの標準ログ機能が搭載されています。Cisco WAE は、*<wae-run-time>*/logs ディレクトリの複数のログファイルにログを記録します。

LDAP 認証ログは、*[wae-run-time]/logs/wae-ldap-auth.log* ファイルに記録されます。*[wae-install-dir]lib/exec/test-java-ssl-conn*にあるツールは、証明書の問題のデバッグに役立つ情報を提供する LDAP 認証や EPNM 通知など、Java アプリケーションの SSL 接続をテストするのに便利です。

最も有用なログは *<wae-run-time>*/logs/wae-java-vm.log です。ほとんどの Cisco WAE パッケージは、このファイルにログを記録します。一部の Cisco WAE パッケージは、*<wae-run-time>*/logs/wae-python-vm-*<package-name>*.log にログを記録します。次の例は、Python-VM ベースのログを示しています。

```
[wae@wae logs]$ pwd
/home/wae/wae-run/logs
[wae@host logs]$ ls -ltr wae-python-vm*
-rw-rw-r-- 1 wae wae    0 Feb 26 07:50 wae-python-vm-cisco-wae-opm-tte.log
-rw-rw-r-- 1 wae wae    0 Feb 26 07:50 wae-python-vm-cisco-wae-get-plan.log
-rw-rw-r-- 1 wae wae    0 Feb 26 07:50 wae-python-vm-cisco-wae-dmdmesh-creator-nimo.log
-rw-rw-r-- 1 wae wae    0 Feb 26 07:50 wae-python-vm-cisco-wae-layout-nimo.log
-rw-rw-r-- 1 wae wae    0 Feb 26 07:50 wae-python-vm-cisco-wae-opm-load-plan.log
-rw-rw-r-- 1 wae wae    0 Feb 26 07:50 wae-python-vm-cisco-wae-dmddeduct-nimo.log
-rw-rw-r-- 1 wae wae    0 Feb 26 07:50 wae-python-vm-cisco-wae-archive.log
-rw-rw-r-- 1 wae wae 2238 Feb 26 07:50 wae-python-vm.log
-rw-rw-r-- 1 wae wae  270 Feb 26 08:20 wae-python-vm-nso_wae_nodes_insert.log
```

デフォルトで、ログレベルは [情報（INFO）] に設定されています。次の方法でロギングを設定できます。

- ランタイムディレクトリの wae.conf ファイルで、さまざまなログのログレベルを定義します。wae.conf ファイルについては、『*Cisco WAE* ユーザーガイド』を参照してください。
- エキスパート モード を使用して、一部のネットワーク インターフェイス モジュール（NIMO）のログ機能を設定します。たとえば、トポロジNIMOやlsp-snmp-nimo モジュールなどのロギング機能を設定できます。エキスパート モードについては、Cisco WAE ユーザー ガイド を参照してください。
- Cisco WAE CLI を使用して、さまざまな NIMO コンポーネントのログレベルを定義します。ログレベルを定義するには、コマンドラインで次のコマンドを入力します。

  ```
  admin@wae% set java-vm java-logging logger <nimo-component> level <level-x>
  ```

  レベルタイプは、level-info、level-debug、level-all です。ログは wae-java-vm.log に保存され、トラブルシューティングに使用できます。

  次の表に、基本的な NIMO コンポーネントを示します。

NIMO コンポーネント	説明
com.cisco.wae	一般的なデバッグ
com.cisco.wae.nimo.topo	トポロジベースの NIMO デバッグ
com.cisco.wae.nimo.lspconfig	NED デバッグによる LSP 設定

NIMO コンポーネント	説明
com.cisco.wae.nimo.lsp	LSP デバッグ
com.cisco.wae.nimo.snmptrafficpoller	SNMP トラフィックポーラーのデバッグ
com.cisco.wae.dare	集約のデバッグ
com.cisco.wae.nimo.optical	オプティカル NIMO デバッグ

ssh: symbol lookup error

waerc ファイルが送信元である場合、waerc によって LD_LIBRARY_PATH 環境変数に設定された WAE 固有の openssl ライブラリが原因で、ssh および scp コマンドが失敗することがあります。エラーメッセージ「`ssh: symbol lookup error: /lib64/libk5crypto.so.3: undefined symbol: Camellia_cbc_encrypt, version OPENSSL_1_1_0`」が表示されます。

この問題を解決するには、次の手順のいずれかを実行します。

- ssh、scp、または waerc が送信元に指定されていないターミナルセッションで 、openssl を使用するその他の操作を使用します。
- waerc を送信元に設定した後に ssh または scp コマンドを使用している場合は、LD_LIBRARY_PATH 環境変数を waerc を送信元に設定する前の値に設定します。

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