使用する前に

ASR 5500 の設定

システムハードウェアが正常にインストールされたら、一連のソフトウェアパラメータを設定し、システムがリロードされるたびに起動されるシステム設定ファイルにこれらの設定を保存する必要があります。

システムに初めて電源が投入されると、アクティブな管理入出力（MIO/UMIO/）カード（通常はシャーシのスロット 5 に取り付けられたカード）が自動的にコンソールポートでクイックセットアップ ウィザードを起動します。このウィザードの指示に従って、システムの初期設定を行います。

シリアルコンソールポート（論理ポート 3）は、MIO カードの前面パネルにあります。

ウィザードを使用せず、コマンドライン インターフェイス（CLI）からコマンドを発行して初期設定を実行することもできます。Exec モードで **setup** コマンドを実行することで、ウィザードを手動で起動できます。詳細については、『*Command Line Interface Reference*』を参照してください。

ASR 5500 クイック セットアップ ウィザードの使用

クイック セットアップ ウィザードは、次の 3 つのパートで構成されています。

- コンテキストレベルのセキュリティ管理者とホスト名の設定
- アウトオブバンド（OOB）管理用のイーサネット インターフェイスの設定
- リモート CLI アクセス用のシステムの設定

クイック セットアップ ウィザード

クイック セットアップ ウィザードは、次の質問に進む前に入力を求める一連の質問で構成されています。一部のプロンプトは、以前の回答によって、または StarOS リリースで特定の機能がサポートされているかどうかによってスキップされる場合があります。

次に、質問のほとんどが表示されるように設計されたクイック セットアップ ウィザードの例を示します（回答を含む）。

```
[local]<host_name># setup
1. Do you wish to continue with the Quick Setup Wizard[yes/no]: yes
2. Enable basic configuration[yes/no]: yes
3. Change chassis key value - WARNING: old configuration scripts will become
invalid after key change[yes/no]: no
5. Create new tech-support password[yes/no]: yes
6. New tech-support password: <ts_password>
7. local context administrator username[admin]: <admin_name>
8. local context administrator password: <admin_password>
9. confirm local context administrator password: <admin_password>
10. hostname[<host_name>]: <host_name>
11. Create single dedicated LI context[yes/no]: no
13. Enable segregated LI configuration[yes/no]: yes
14. Enable LOCAL interface[yes/no]: yes
17. LOCAL Out of band Ip Address: <ip_address>
18. LOCAL Out of band subnet mask: <subnet_mask>
19. Default gateway Ip Address: <gw_ip_address>
20. Enable remote access[yes/no]: yes
21. Enable sshd[yes/no]: yes
22. Enter a default SSH key size[2048/3072/4096/5120/7168/9216]: 2048
23. Enable sftp server[yes/no]: yes
24. Enable telnetd[yes/no]: no
25. Enable ftpd[yes/no]: no
Do you want to review your selections[no/yes]: no
Do you want to view the configuration script created[yes/no]: yes
<configuration_script_output>
Do you want to apply configuration script created[yes/no]: no
[local]<host_name>#
```

表 *1:* クイック セットアップ ウィザードの質問

質問	タスク	説明/注意事項
1	ウィザードを開始または終了します。	プロンプトで **no** と入力すると、コマンドライン インターフェイス（CLI）が自動的に表示されます。CLI を使用してシステムの初期設定を行う手順については、CLI を使用した初期設定（12 ページ）に進みます。 コマンドプロンプトで **setup** と入力して、ウィザードを再起動します。
2	基本設定を有効にします。	基本設定ファイルを作成するには、**yes** と入力します。
3	シャーシのキー値を変更します。	工場出荷時にシステムごとに固有のシャーシキーが設定されています。このキーは、生成された設定ファイルにある暗号化されたパスワードを復号するために使用されます。システム管理者は、設定ファイルに保存されているパスワードの暗号化に使用される固有のシャーシキーを作成できます。 **yes** と入力して、新しいシャーシキーを設定します。詳細については、「システム設定」の手順を参照してください。詳細については、「システムセキュリティ」の章を参照してください。
5、6	tech-support のパスワードを作成します。	詳細については、「システムセキュリティ」の章の「*CLI* テストコマンドにアクセスのためのパスワードの有効化」を参照してください。
7	システムの管理ユーザー名を設定します。	ウィザードを使用して設定されたデフォルトの管理ユーザーの名前は *admin* です。 管理ユーザー名は、大文字と小文字を区別した 1 ～ 32 文字の英数字の文字列です。
8、9	システムの管理パスワードを設定します。	管理ユーザーのパスワードは、大文字と小文字を区別した 1 ～ 63 文字の英数字の文字列です。リリース 21.0 以降では、パスワードとして 127 文字入力できます。
10	システムのホスト名を変更します。	StarOS CLI プロンプトにホスト名が 表示されます。

質問	タスク	説明/注意事項
11	単一の専用 LI コンテキストを作成します。	専用 LI コンテキストを作成する前に、『*Lawful Intercept Configuration Guide*』を参照してください。作成後は、専用 LI コンテキストを元に戻すことはできません。
13	分離した LI 設定を有効にします。	システム設定と LI の設定を分離する前に、『*Lawful Intercept Configuration Guide*』を参照してください。
14, 17, 18	アウトオブバンドシステムの管理用に 1 つの入出力（MIO/UMIO）アウトオブバンド管理インターフェイスを設定します。	管理 LAN 上のトラフィックは、ユーザーデータおよび制御シグナリングと同じメディアを介して転送されません。 セキュリティ上の理由から、管理機能をユーザーデータと制御シグナリングから分離したネットワーク上で維持することを推奨します。 MIO ポート 1（mio1）は、1000Base-T のデフォルトの管理ポートです。 MIO ポート 2（mio2）は、セカンダリ管理ポートとして使用できます。 システムを CAT5 イーサネットケーブルを使用して管理ネットワークに接続するには、RJ-45 インターフェイスを使用します。 インターフェイスに IP アドレスとサブネットマスクを設定します。
19	インターフェイスごとにデフォルトゲートウェイを設定します。	IP アドレスを入力します。
20	リモートアクセスを有効にします。	yes と入力し、このシステムへのリモートアクセスを許可します。 MIO 上に 2 番目の管理インターフェイスを設定する手順については、「システム設定」の章を参照してください。

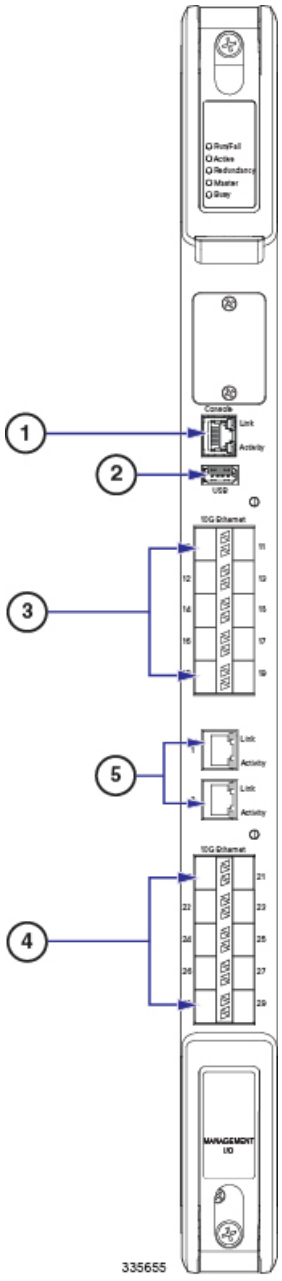
質問	タスク	説明/注意事項
21 ～ 23	システムにアクセスするための SSH リモートアクセスプロトコルを有効にします。	セキュアシェル（SSH）は、デフォルトでTCP ポート番号 22 を使用します（有効になっている場合）。 SSH キーのサイズを指定できます。SSH v2-RSA のキー生成では、そのキーサイズの値が使用されます。
		注：セキュリティを最大限にするには、SSH v2 のみを使用してください。 SSH v2 のみがサポートされています。
		Secure File Transfer Protocol（SFTP）は、デフォルトで TCP ポート番号 22を使用します（有効になっている場合（subsystem sftp））。
24	Telnet経由でリモートアクセスを有効にします。	**注**：システムセキュリティを最大限にするには、telnet プロトコルを有効にしないでください。 **注**：Telnet はサポートされていません。
25	システムへの FTP アクセスを有効にします。	デフォルトでは、File Transfer Protocol（FTP）は TCP ポート番号 21 を使用します（有効になっている場合）。 **注**：システムセキュリティを最大限にするには、FTPを有効にしないでください。 **注**：FTP はサポートされていません。
—	前のプロンプトの設定の確認や変更を行います。	**1.** 変更するプロンプトの番号を入力します。 **2.** パラメータを設定します。 **3.** オプションです。追加の設定を変更するには、ステップ *1* とステップ *2* を繰り返します。 **4.** すべての変更を完了したら、「done」と入力します。
—	入力に基づいてウィザードによって作成されたスクリプトの設定を確認します。	作成したスクリプトの例を下記の例に示します。変数はイタリック体で表示されます（*variable*）。

質問	タスク	説明/注意事項
—	設定ファイルをシステムに適用します。	適用されると、パラメータ設定は、MIO/UMIO のフラッシュメモリに保存されている system.cfg ファイルに自動的に保存されます。

```
Do you want to view the configuration script created[yes/no]: y
config
   system hostname hostname
   context local
      administrator admin_name password passwd
      interface mio1
         ip address ip_address subnet
         #exit
      ip route 0.0.0.0 0.0.0.0 gw_address mio1
      ssh key v1_key
      ssh key v2_rsa_key
      ssh key v2_dsa_key
      server sshd
      subsystem sftp
      #exit
      no server telnetd
      no server ftpd
      #exit
      port ethernet 5/1
      bind interface mio1 local
      no shutdown
   #exit
end
Do you want to apply configuration script created[yes/no]:
```

重要 ウィザードを使用した設定が完了したら、他のシステムパラメータの設定方法に関する手順に進みます。

図 *1 : MI0* インターフェイス



1	コンソールポート（ポート 3）	2	USB ポート
3	10 GbE ポート、DC-1（ポート 10 ～ 19）	4	10 GbE ポート、DC-2（ポート 20 ～ 29）
5	1 GbE ポート（1000Base-T）（ポート 1 と 2）		

StarOS クイック セットアップ ウィザードの使用

クイック セットアップ ウィザードは、次の 3 つのパートで構成されています。

- コンテキストレベルのセキュリティ管理者とホスト名の設定
- アウトオブバンド（OOB）管理用のイーサネット インターフェイスの設定
- Telnet、セキュアシェル（SSH）、または File Transfer Protocol（FTP）を介したリモート CLI アクセスのためのシステム設定

次の図表は、ウィザードの実行ロジックを追加情報と注意事項とともに示すフロー図です。

図 *2 : StarOS* クイック セットアップ ウィザードのロジック図

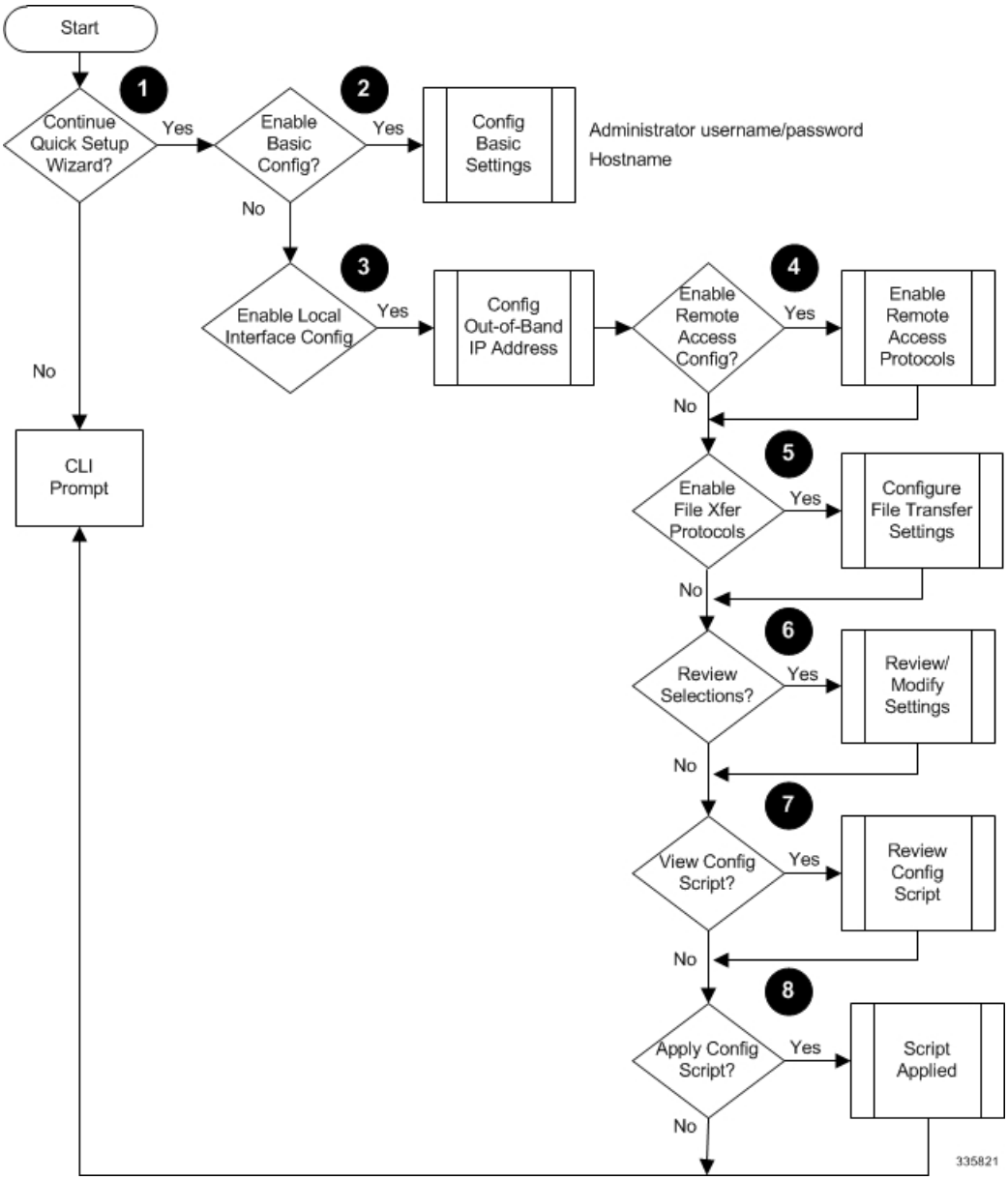


表 *2:* クイック セットアップ ウィザードのロジック図に関する説明

項目	タスク	説明/注意事項
1	ウィザードを開始または終了します。	プロンプトで **no** と入力すると、コマンドラインインターフェイス（CLI）が自動的に表示されます。CLI を使用してシステムの初期設定を行う手順については、StarOS CLI を使用した初期設定 （14 ページ）に進みます。
		コマンドプロンプトで **setup** と入力して、ウィザードを再起動します。
2	リリース *19.2* 以降の場合は、tech-support のパスワードを設定します。	tech-support のパスワードは、cli test-commands にアクセスするために使用されます。
	システムの管理ユーザー名/パスワードおよびホスト名を設定します。	ウィザードを使用して設定されたデフォルトの管理ユーザーの名前は *admin* です。
		管理ユーザーの名前は、大文字と小文字を区別した 1 ～ 32 文字の英数字の文字列です。
		管理ユーザーのパスワードは、大文字と小文字を区別した 1 ～ 63 文字の英数字の文字列です。リリース 21.0 以降では、パスワードとして 127 文字入力できます。
		NULL 以外の有効なホスト名を設定します。ホスト名は、大文字と小文字が区別される 1 ～ 63 文字の英数字文字列です。
3	システムに対して、単一のアウトオブバンド管理インターフェイスを設定します。	管理 LAN 上のトラフィックは、ユーザーデータおよび制御シグナリングと同じメディアを介して転送されません。
		セキュリティ上の理由から、管理機能をユーザーデータと制御シグナリングから分離したネットワーク上で維持することを推奨します。
		インターフェイスの IP アドレス、サブネットマスク、およびゲートウェイを設定します。

項目	タスク	説明/注意事項
4	システムにアクセスするためのリモートアクセスプロトコルを有効にします。	セキュアシェル（SSH）は、デフォルトでTCポート番号 22 を使用します（有効になってる場合）。 SSH V1 や V2 がサポートされています。 **セキュリティを最大限にするには、SSH v2 みを使用してください。**
5	システムとの間でファイルをコピーするためのファイル転送プロトコルを有効にします。	SSH が有効になっている場合は、SSH ファイ転送プロトコル（SFTP）サーバー機能を有にすることもできます。 デフォルトでは、Telnet は TCP ポート番号 2を使用します（有効になっている場合）。 デフォルトでは、SSH File Transfer Protocol（FTP）は TCP ポート番号 21 を使用します（効になっている場合）。
6	前のプロンプトの設定の確認や変更を行います。	**1.** 変更するプロンプトの番号を入力します **2.** パラメータを設定します。 **3.** オプションです。追加の設定を変更するは、ステップ *1* とステップ *2* を繰り返しす。 **4.** すべての変更を完了したら、「done」と力します。
7	入力に基づいてウィザードによって作成されたスクリプトの設定を確認します。	次の例では、作成されたスクリプトのサンプが表示されています。変数はイタリック体で示されます（*variable*）。
8	設定ファイルをシステムに適用します。	適用されると、パラメータ設定は、フラッシメモリに保存されている system.cfg ファイル自動的に保存されます。

クイック セットアップ スクリプトの例

EXEC モードの **setup** コマンドを実行すると、次のようにセットアップスクリプトが表示されます。

```
[local]host_name# setup
1. Do you wish to continue with the Quick Setup Wizard[yes/no]: yes
2. Enable basic configuration[yes/no]: yes
3. Change chassis key value[yes/no]: yes
4. New chassis key value: chassis_key
```

```
5. Create new tech-support password[yes/no]: yes
6. New tech-support password: new_password
7. local context administrator username[admin]: context_admin_username
8. local context administrator password: context_admin_password
9. confirm local context administrator password: context_admin_password
10. hostname[asr5500]: hostname
```

CLI を使用した初期設定

初期設定は次のように構成されています。

- コンテキストレベルのセキュリティ管理者とホスト名の設定
- MIO/UMIO カード上のイーサネット インターフェイスの設定
- Telnet、SSH、または FTP（セキュアまたは非セキュア）によるリモート CLI アクセスのためのシステム設定

重要 FTP および telnet はサポートされていません。

この項では、CLI を使用してこれらのタスクを実行するための手順を説明します。

手順

ステップ 1 CLI プロンプトで、次のように入力します。

```
[local]host_name# configure
[local]host_name(config)#
```

ステップ 2 次のコマンドを入力してコンテキスト構成モードを開始します。

```
[local]host_name(config)# context local
[local]host_name(config-ctx)#
```

ローカルコンテキストは、システムの管理コンテキストです。コンテキストを使用すると、サービスまたはインターフェイスを論理的にグループ化することができます。1 つのコンテキストは複数のサービスで構成でき、複数のインターフェイスにバインドできます。

ステップ 3 システムのコンテキストレベルのセキュリティ管理者、設定管理者、オペレータ、およびインスペクタを設定するには、次のコマンドを入力します。

administrator *user_name* **[encrypted] [nopassword] password** *password* **[max-age** *days*]**[no-max-age]| [ecs] [expiry-date** *date_time* **] [ftp [sftp-server** *sftp_name* **]] [li-administration] [nocli] [noconsole] [noecs] [timeout-absolute** *timeout_absolute* **] [timeout-min-absolute** *timeout_min_absolute* **] [timeout-idle** *timeout_idle* **][timeout-min-idle** *timeout_min_idle* **][exp-grace-interval** *days*]**[exp-warn-interval** *days*]**[no-exp-grace-interval][no-exp-warn-interval]**

no administrator *user_name*

初期設定時にコンテキストレベルのセキュリティ管理者を設定する必要があります。初期設定プロセスが完了し、CLI セッションを終了した後、セキュリティ管理者が設定されていない場合は、CLI アクセスがロックされます。この項のコマンドの詳細については、『*Command Line Interface Reference*』の「*Context Configuration Mode Commands*」の章を参照してください。

（注） セキュリティ上の理由から、**li-administration** アカウントは、一般的なシステム管理ではなく、合法的傍受（LI）機能でのみ使用するよう制限する必要があります。セキュリティ管理者と管理者のみが、LI 権限をプロビジョニングできます。司法当局（LEA）の基準に従ってセキュリティを確保するために、LI 管理ユーザーは、セキュアシェル（SSH）プロトコルのみを使用してシステムにアクセスする必要があります。LI 権限は、システム全体の単一コンテキスト内で使用するように設定することもできます。詳細については、『*Lawful Intercept Configuration Guide*』を参照してください。

ステップ 4 プロンプトで次のコマンドを入力して、コンテキストの構成モードを終了します。

```
[local]host_name(config-ctx)# exit
[local]host_name(config)#
```

ステップ 5 次のコマンドを入力して、システムがネットワーク上で認識されるホスト名を設定します。

```
[local]host_name(config)# system hostname host_name
```

host_name は、ネットワーク上でシステムが認識される名前です。ホスト名は、大文字と小文字が区別される 1 ～ 63 文字の英数字文字列です。

ステップ 6 次の手順を使用して、MIO/UMIO のネットワークインターフェイスを設定します。

a) 次のコマンドを入力して、コンテキスト構成モードを開始します。

```
[local]host_name(config)# context local
[local]host_name(config-ctx)#
```

b) インターフェイスの名前を指定するには、次のコマンドを入力します。

```
[local]host_name(config-ctx)# interface interface_name
```

interface_name は、大文字と小文字が区別される 1 ～ 79 文字の英数字の文字列で表されるインターフェイスの名前です。システムがイーサネット インターフェイスの構成モードを開始する、次のプロンプトが表示されます。

```
[local]host_name(config-if-eth)#
```

c) 次のコマンドを入力して、前のステップで設定したインターフェイスの IP アドレスを設定します。

{ ip address | ipv6 address } *ipaddress subnetmask*

クイック セットアップ ウィザードで誤って設定されたアドレスまたはサブネットを修正するためにこのコマンドを実行する場合は、デフォルトルートとポートバインドの設定を確認する必要があります。この手順のステップ *11* とステップ *6* を使用します。問題がある場合は、ステップ *7e* ～ *7k* を実行して情報を再設定します。

d) 次のコマンドを入力して、イーサネット インターフェイスの構成モードを終了します。

```
[local]host_name(config-if-eth)# exit
[local]host_name(config-ctx)#
```

e) 必要に応じてスタティックルートを設定して、システムをデフォルトゲートウェイに指定します。次のコマンドを入力します。

{ ip | ipv6 } route *gw_address interface_name*

f) コンテキストの構成モードを終了するには、次のように入力します。

```
[local]host_name(config-ctx)# exit
[local]host_name(config)#
```

g) イーサネットポートの構成モードを開始します。

port ethernet *slot#/port#*

h) ステップ 7b で作成したインターフェイスにポートをバインドします。バインドにより、ポートとそのすべての設定がインターフェイスに関連付けられます。次のコマンドを入力します。

```
[local]host_name(config-port-<slot#/port#>)# bind interface interface_name local
[local]host_name(config-port-<slot#/port#>)# no shutdown
```

*interface_name*は、ステップ *7b* で設定したインターフェイスの名前です。

i) 次のコマンドを入力して、イーサネット インターフェイスの構成モードを終了します。

```
[local]host_name(config-port-<slot#/port#>)# exit
[local]host_name(config)#
```

重要 2 番目の IP アドレスを使用して MIO/UMIO 管理インターフェイスを設定する手順については、次を参照してください。

StarOS CLI を使用した初期設定

初期設定は次のように構成されています。

- コンテキストレベルのセキュリティ管理者とホスト名の設定
- vNIC でのイーサネット インターフェイスの設定
- Telnet、SSH、または FTP（セキュアまたは非セキュア）によるリモート CLI アクセスのためのシステム設定

この項では、CLI を使用してこれらのタスクを実行するための手順を説明します。

手順

ステップ 1 ハイパーバイザを介してコンソールポートにログインしてください。

ステップ 2 CLI プロンプトで、次のように入力します。

```
[local]host_name configure[local]host_name(config)
```

ステップ 3 次のコマンドを入力してコンテキスト構成モードを開始します。

```
[local]host_name(config) context local[local]host_name(config-ctx)
```

ローカルコンテキストは、システムの管理コンテキストです。コンテキストを使用すると、サービスまたはインターフェイスを論理的にグループ化することができます。1 つのコンテキストは複数のサービスで構成でき、複数のインターフェイスにバインドできます。

ステップ 4 システムのコンテキストレベルのセキュリティ管理者を設定するには、次のコマンドを入力します。

administrator *user_name* **[encrypted] password** *password* **| [ecs] [expiry-date** *date_time* **] [ftp] [li-administration] [nocli] [noecs]]**

（注） 初期設定時にコンテキストレベルのセキュリティ管理者を設定する必要があります。初期設定プロセスが完了し、CLI セッションを終了した後、セキュリティ管理者が設定されていない場合は、CLI アクセスがロックされます。このコマンドの詳細については、『*Command Line Interface Reference*』の「*Context Configuration Mode Commands*」の章を参照してください。

ステップ 5 プロンプトで次のコマンドを入力して、コンテキストの構成モードを終了します。

```
[local]host_name(config-ctx) exit
[local]host_name(config)
```

ステップ 6 次のコマンドを入力して、システムがネットワーク上で認識されるホスト名を設定します。

```
[local]host_name(config) system hostname host_name
```

host_name は、ネットワーク上でシステムが認識される名前です。ホスト名は、大文字と小文字が区別される 1 ～ 63 文字の英数字文字列です。デフォルトのホスト名は「qvpc-si」です。

ステップ 7 vNIC 上のネットワーク インターフェイスを次のように設定します。

a) 次のコマンドを入力して、コンテキスト構成モードを開始します。

```
[local]host_name(config) context local
[local]host_name(config-ctx)
```

b) インターフェイスの名前を指定するには、次のコマンドを入力します。

```
[local]host_name(config-ctx) interface interface_name
```

interface_name は、大文字と小文字が区別される 1 ～ 79 文字の英数字の文字列で表されるインターフェイスの名前です。システムがイーサネット インターフェイスの構成モードを開始する、次のプロンプトが表示されます。

```
[local]host_name(config-if-eth)
```

c) 次のコマンドを入力して、前のステップで設定したインターフェイスの IP アドレスを設定します。

{ ip address | ipv6 address } *ipaddress subnetmask*

（注） クイックセットアップ ウィザードで誤って設定されたアドレスまたはサブネットを修正するためにこのコマンドを実行する場合は、デフォルトルートとポートバインドの設定を確認する必要があります。この手順のステップ *11* とステップ *6* を使用します。問題がある場合は、ステップ *7e* ～ *7k* を実行して情報を再設定します。

d) 次のコマンドを入力して、イーサネット インターフェイスの構成モードを終了します。

```
[local]host_name(config-if-eth) exit
[local]host_name(config-ctx)
```

e) 必要に応じてスタティックルートを設定して、システムをデフォルトゲートウェイに指定します。次のコマンドを入力します。

```
{ ip | ipv6 } route gw_address interface_name
```

f) コンテキストの構成モードを終了するには、次のように入力します。

```
[local]host_name(config-ctx) exit
[local]host_name(config)
```

g) イーサネットポートの構成モードを開始します。

```
port ethernet slot/port
```

VPC の場合、スロット番号は常に「1」です。vNIC トラフィックポートは 10 ～ 21 です。ポート 1 は管理ポートです。

h) ステップ 7b で作成したインターフェイスにポートをバインドします。バインドにより、ポートとそのすべての設定がインターフェイスに関連付けられます。次のコマンドを入力します。

```
[local]host_name(config-port-slot/port) bind interface interface_name local
[local]host_name(config-port-slot/port) no shutdown
```

*interface_name*は、ステップ *7b* で設定したインターフェイスの名前です。

i) 次のコマンドを入力して、イーサネット インターフェイスの構成モードを終了します。

```
[local]host_name(config-port-slot/port) exit
[local]host_name(config)
```

（注） 管理ポートは、VLAN もサポートしています。詳細については、「インターフェイスとポート」の章の「*VLAN*」の項を参照してください。

2 番目の IP アドレスを使用して vNIC 管理インターフェイスを設定する手順については、以下を参照してください。

システム管理ユーザーの設定

この項では、セキュリティ管理者がユーザーアカウントを制御できるようにするセキュリティ機能の一部について説明します。

同時 CLI セッション数の制限

セキュリティ管理者は同時対話型CLIセッションの数を制限できます。同時対話型セッションの数を制限すると、システム全体のリソースの消費量が削減されます。また、ユーザーがすでに使用されている機密ユーザー情報にアクセスする可能性を防ぎます。

ほとんどの特権アカウントでは、複数の同時ログインは必要ありません。

（注） 21.9 以降のリリースでは、1 つの CLI セッションでの複数のチャネルはサポートされていません。

重要 すべての特権アカウントには、セッションの最大数を設定することを推奨します。

セキュリティ管理者は、その特定のユーザーアカウントに使用される認証方式に応じて、3 つの異なる方法で同時インタラクティブ CLI セッションの数を制限できます。

StarOS は次の 3 つのログイン認証方式をサポートしています。

- TACACS+ サーバーユーザー
- ローカルユーザーのユーザー
- AAA コンテキストユーザー

TACACS+ サーバーユーザーの最大セッション数の設定の詳細については「動作」を参照してください。ローカルユーザーのユーザーと AAA コンテキストユーザーの最大セッション数の設定の詳細については「Configuring Context-Level Administrative Users」を参照してください。

各認証方式は、3 つの認証方式のそれぞれが同じユーザー名を使用できるため、個別に設定する必要があります。

CLI セッションの自動ログアウト

セキュリティ管理者は、特定のユーザーアカウントの自動ログアウトを設定できます。対話型 CLI セッションが使用可能な時間を分単位で制限すると、システム全体のリソースの消費量が削減されます。また、アイドル状態のままになっている端末ウィンドウで、ユーザーがユーザーアカウントにアクセスする可能性を防ぐこともできます。この項で説明されているすべての認証方式は、アイドル セッション タイムアウトの手法と絶対セッションタイムアウトの手法の両方をサポートしています。

ほとんどの特権アカウントは、無期限のログインタイムアウトの制限を必要としません。

重要 すべての特権アカウントには、セッションタイムアウトを設定することを推奨します。

show tacacs summary コマンドと **show tacacs session id** コマンドのアイドルタイムアウトおよびセッションタイムアウトのフィールドを使用すると、管理者は特定のアカウントの自動ログアウトを設定できます。

セッションタイムアウト：セキュリティ管理者は、セッションが自動的に切断される前に、ユーザーがセッションにログオンできる最大時間を分単位で指定できます。

アイドルタイムアウト：セキュリティ管理者は、セッションが自動的に切断される前に、セッションがアイドル状態を維持できる最大時間を分単位で指定できます。

重要 セッションタイムアウトとアイドルタイムアウトのフィールドは排他的ではありません。両方が指定されている場合は、低いセッションタイムアウトが常に最初に到達するため、アイドルタイムアウトは常にセッションタイムアウトよりも低くする必要があります。

対話型CLIセッションを使用できる最大時間を分単位で設定する方法の詳細については、『*CLI Reference*』の **dle-sessions threshold** コマンドと **clear tacacs sessions** CLI コマンド、および『*Statistics and Counter Reference*』の **show tacacs summary** と **show tacacs session id** を参照してください。

リモートアクセス用のシステムの設定

リモートアクセス用にシステムを設定します。管理ユーザーは、ローカルエリアネットワーク（LAN）またはワイドエリアネットワーク（WAN）を介して、リモートロケーションからシステムにアクセスできます。

- Telnet
- セキュア シェル（SSH）
- File Transfer Protocol（FTP）（セキュアまたは非セキュア）
- Trivial File Transfer Protocol（TFTP）

重要 2つの同時telnetセッションがあり、1人の管理者が他の管理者がログに記録するコンテキストを削除した場合は、削除されたコンテキストの管理者が自動的にローカルコンテキストに退出させられることはありません。削除されたコンテキストはCLIプロンプトに引き続き表示されますが、コンテキスト固有のコマンドによってエラーが生成されます。

重要 セキュリティを最大限にするには、SSH v2 を使用します。

重要 FTP および telnet はサポートされていません。

手順

ステップ 1 次のコマンドを入力してコンテキスト構成モードを開始します。

```
[local]host_name(config)# context local
[local]host_name(config-ctx)#
```

ステップ 2 必要に応じて、Telnet アクセスを許可するようにシステムを設定します。

```
[local]host_name(config-ctx)# server telnetd
```

システムセキュリティを最大限にするには、Telnet を有効にしないでください。

ステップ 3 SSH アクセスを許可するようにシステムを設定します。

```
[local]host_name(config-ctx)# ssh generate key [ type { v2-rsa | v2-dsa } ]
```

v2-rsa は推奨されるキータイプです。

v2-dsa キーワードは、コンテキスト構成モードの **ssh generate** CLI コマンド内に隠されています。以前のリリースでサポートされていたキーワードが後続のリリースでは隠されている可能性があります。StarOS は、以前のリリースで作成された既存のスクリプトと構成ファイル内の隠されたキーワードを引き続き解析します。ただし、新しいスクリプトや構成ファイルで使用するために、コマンドシンタックスに隠されたキーワードは表示されなくなりました。疑問符（?）を入力しても、ヘルプテキストの一部として隠しキーワードは表示されません。削除されたキーワードを指定すると、解析時にエラーメッセージが生成されます。

```
[local]host_name(config-ctx)# ssh generate key type v2-rsa
```

ステップ 4 SFTP をサポートするようにシステムを設定します。

```
[local]host_name(config-ctx)# server sshd
[local]host_name(config-sshd)# subsystem sftp
[local]host_name(config-sshd)# exit
```

SSH の詳細については、SSH オプションの設定 （23 ページ）を参照してください。

ステップ 5 必要に応じて、次のコマンドを入力して、FTP アクセスを許可するようにシステムを設定します。

```
[local]host_name(config-ctx)# server ftpd
```

システムセキュリティを最大限にするために、 FTP を有効にしないでください。これはサポートされていません。

ステップ 6 必要に応じて、TFTP アクセスを許可するようにシステムを設定します。

```
[local]host_name(config-ctx)# server tftpd
```

ステップ 7 次のコマンドを入力して、構成モードを終了します。

```
[local]host_name(config-ctx)# end
[local]host_name#
```

ステップ 8 次のコマンドを入力して、設定を確認します。

```
[local]host_name# show configuration
```

CLI 出力は、次の出力例のようになります。

```
context local
   interface interface_name
      ip address ipaddress subnetmask
      exit
   subscriber default
      exit
   administrator admin_name password admin_password
   no server telnetd
   no server ftpd
   ssh generate key
   server sshd
   subsystem sftp
   exit
port ethernet 5/1
   bind interface interface_name local
   exit
port ethernet 5/1
   no shutdown
   exit
 snmp engine-id local 800007e580ed826c191ded2d3d
end
```

ステップ 9 次のコマンドを入力して、IP ルートの設定を確認します。

```
[local]host_name# show ip route
```

CLI 出力は、次の出力例のようになります。

```
"*" indicates the Best or Used route.
 Destination         Nexthop          Protocol   Prec Cost Interface
*0.0.0.0/0           ipaddress      static     1    0    mio1
*network           0.0.0.0          connected  0    0    mio1
```

ステップ 10 次のコマンドを入力して、インターフェイス バインディングを確認します。

```
[local]host_name#  show ip interface name interface_name
```

nterface_name> は、ステップ *7b* で設定されたインターフェイスの名前です。CLI 出力は、出力例のようになります。

```
Intf Name:                      mio1
Intf Type:                      Broadcast
Description:
IP State:                       UP (Bound to 5/1 untagged, ifIndex 83951617)
IP Address:                     ipaddress      Subnet Mask:    subnetmask
Bcast Address:                  bcastaddress   MTU:            1500
Resoln Type:                    ARP            ARP timeout:        3600 secs
Number of Secondary Addresses:  0
```

ステップ 11 「設定の確認と保存」の説明に従って、設定を保存します。

リモートアクセス用のシステムの設定

リモートアクセス用にシステムを設定します。管理ユーザーは、管理ネットワークを介してリモートの場所からインスタンスにアクセスできます。

- Telnet
- セキュア シェル（SSH）
- File Transfer Protocol（FTP）（セキュアまたは非セキュア）
- Trivial File Transfer Protocol（TFTP）

（注） 2つの同時telnetセッションがあり、1人の管理者が他の管理者がログに記録するコンテキストを削除した場合は、削除されたコンテキストの管理者が自動的にローカルコンテキストに退出させられることはありません。削除されたコンテキストはCLIプロンプトに引き続き表示されますが、コンテキスト固有のコマンドによってエラーが生成されます。

（注） セキュリティを最大限にするには、SSH v2 を使用します。

（注） FTP および telnet はサポートされていません。

手順

ステップ 1 次のコマンドを入力してコンテキスト構成モードを開始します。

```
[local]cf_host_name(config) context local
[local]cf_host_name(config-ctx)
```

ステップ 2 必要に応じて、Telnet アクセスを許可するようにシステムを設定します。

```
[local]cf_host_name(config-ctx) server telnetd
```

ステップ 3 必要に応じて、SSH アクセスを許可するようにシステムを設定します。

```
[local]cf_host_name(config-ctx) ssh generate key [ type v2-rsa ]
```

（注） **v2-rsa**は推奨されるキータイプです。

（注） **v2-dsa** キーワードは コンテキスト構成モードの **ssh generate** CLI コマンド内に隠されています。以前のリリースでサポートされていたキーワードが後続のリリースでは隠されている可能性があります。システムは、以前のリリースで作成された既存のスクリプトや設定ファイル内の隠されたキーワードを引き続き解析します。ただし、新しいスクリプトや構成ファイルで使用するために、コマンドシンタックスに隠されたキーワードは表示されなくなりました。疑問符（?）を入力しても、ヘルプテキストの一部として隠しキーワードは表示されません。削除されたキーワードを指定すると、解析時にエラーメッセージが生成されます。

```
[local]cf_host_name(config-ctx) server sshd
[local]cf_host_name(config-sshd) subsystem sftp
[local]cf_host_name(config-sshd) exit
```

ステップ 4 必要に応じて、次のコマンドを入力して、FTP アクセスを許可するようにシステムを設定します。

```
[local]cf_host_name(config-ctx) server ftpd
```

ステップ 5 次のコマンドを入力して、構成モードを終了します。

```
[local]cf_host_name(config-ctx) end
[local]cf_host_name
```

ステップ 6 次のコマンドを入力して、設定を確認します。

```
[local]cf_host_name show configuration
```

CLI 出力は、次の出力例のようになります。

```
context local
   interface interface_name
      ip address ipaddress subnetmask
      exit
   subscriber default
      exit
   administrator admin_name password admin_password
   server telnetd
   server ftpd
   ssh generate key
   server sshd
   subsystem sftp
   exit
port ethernet 1/1
   bind interface interface_name local
   exit
port ethernet 1/1
   no shutdown
   exit
 snmp engine-id local 800007e580ed826c191ded2d3d
end
```

ステップ 7 次のコマンドを入力して、IP ルートの設定を確認します。

```
[local]cf_host_name show ip route
```

CLI 出力は、次の出力例のようになります。

```
"*" indicates the Best or Used route.
 Destination         Nexthop         Protocol  Prec Cost Interface
*0.0.0.0/0           ipaddress      static     1    0    vnic1
*network            0.0.0.0         connected  0    0    vnic1
```

ステップ 8 次のコマンドを入力して、インターフェイス バインディングを確認します。

```
[local]cf_host_name show ip interface name interface_name
```

interface_name は、手順 *7b* で設定したインターフェイスの名前です。CLI 出力は、次の出力例のようになります。

```
Intf Name:       vnic1

Description:
IP State:        UP (Bound to 1/1 untagged, ifIndex 83951617)
IP Address:      ipaddress          Subnet Mask:     subnetmask
Bcast Address:   bcastaddress       MTU:             1500
Resoln Type:     ARP                ARP timeout:     3600 secsL3 monitor LC-port
switchover: DiasabledNumber of Secondary Addresses:  0
```

ステップ 9 「設定の確認と保存」の章の説明に従って、設定を保存します。

SSH オプションの設定

SSHv2 RSA は、StarOS でサポートされる SSH の唯一のバージョンです。SSHv2 DSA は、StarOS CLI 内に隠されています。

重要 以前のリリースでサポートされていたキーワードが後続のリリースでは隠されている可能性があります。StarOS は、以前のリリースで作成された既存のスクリプトと構成ファイル内の隠されたキーワードを引き続き解析します。ただし、新しいスクリプトや構成ファイルで使用するために、コマンドシンタックスに隠されたキーワードは表示されなくなりました。疑問符（?）を入力しても、ヘルプテキストの一部として隠しキーワードは表示されません。削除されたキーワードは、解析時にエラーメッセージを生成します。

SSH プロトコルのバージョン 1 は、セキュリティの脆弱性が原因で廃止されました。**v1-rsa** キーワードは、コンテキスト構成モードの **ssh** コマンドのために削除されました。SSHv1-RSA キーを使用するスクリプトまたは設定を実行すると、エラーメッセージが返され、イベントログが生成されます。次に、エラーメッセージの出力例を示します。

```
CLI print failure Failure: SSH V1 contains multiple structural vulnerabilities and is
no longer considered secure. Therefore we don't support v1-rsa SSH key any longer, please
 generate a new v2-rsa key to replace this old one.
```

v1-rsa キーを含む設定からシステムが起動する場合、SSH を介してログインするときに起動の失敗が予想されます。回避策は、コンソールポートを介してログインし、新しい ssh v2-rsa キーを再生成し、サーバー sshd を設定することです。その後、ssh を介してログインできるようになります。

コンテキスト構成モードの **ssh** コマンドでは、**v2-dsa キーワード**が隠されるようになりました。

v1-rsa キーワードは、Exec モードの **show ssh key** CLI コマンドから削除されました。

SSH ホストキー

SSH キーベースの認証では、誰に対しても表示が許可されている「公開」キーと、所有者のみが表示を許可されている別の「秘密」キーの、2 つのキーを使用します。キーペアを作成し、ログインするデバイスに秘密キーを安全に保存して、ログインするシステム（ASR 5500VPC-SI）に公開キーを保存します。

SSH ホストキーは、指定された StarOS コンテキスト内で生成されます。コンテキストは、ユーザーインターフェイスに関連付けられています。

コンテキストに関連付けられている sshd サーバーにアクセスするための承認されたキーを持つ管理ユーザー名を設定または削除します。

SSH キーのサイズ設定

グローバル構成モードの **ssh key-size** CLI コマンドは、すべてのコンテキストの SSH キー生成のキーサイズを設定します（RSA ホストキーのみ）。

手順

ステップ 1 グローバル構成モードを開始します。

```
[local]host_name# configure
[local]host_name(config)#
```

ステップ 2 SSH キーのビットサイズを指定します。

```
[local]host_name(config)# ssh key-size { 2048 | 3072 | 4096 | 5120 | 6144 | 7168 |
9216 }
```

SSH キーのデフォルトのビットサイズは 2048 ビットです。

SSH キー生成の待機時間の設定

SSH キーは、最後のキー生成以降に設定可能な時間間隔が経過した後にのみ生成できます。**ssh key-gen wait-time** コマンドは、この待機時間を秒単位で指定します。デフォルトの間隔は 300 秒（5 分）です。

手順

ステップ 1 コンテキスト構成モードを開始します。

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

ステップ 2 待機時間間隔を指定します。

```
[local]host_name(config-ctx)# ssh key-gen wait-time seconds
[local]host_name(config-ctx)#
```

注：

- *seconds* を 0 ～ 86400 の整数で指定します。デフォルト = 300

SSH 暗号化暗号方式の指定

SSH 構成モードの **暗号** CLI コマンドは、SSH 対称暗号化のために、sshd の暗号優先順位リストを設定します。そのコンテキストの暗号オプションが変更されます。

手順

ステップ 1 SSH 構成モードを開始します。

```
[local]host_name(config-ctx)# server sshd
```

ステップ 2 必要な暗号化アルゴリズムを指定します。

```
[local]host_name(config-sshd)# ciphers algorithms
```

注：

- アルゴリズムは 1 ～ 511 文字の英数字の文字列で、次に示すように、優先順位（左から右）でカンマ区切りの変数（スペースなし）の単一の文字列として使用するアルゴリズムを指定します。
  - **blowfish-cbc**：対称キーブロック暗号、暗号ブロック連鎖（CBC）
  - **3des-cbc**：トリプルデータ暗号化規格、CBC
  - **aes128-cbc**：Advanced Encryption Standard（AES; 高度暗号化規格）、128 ビットキーサイズ、CBC
  - **aes128-ctr**：AES、128 ビットキーサイズ、カウンタモード暗号化（CTR）
  - **aes192-ctr**：AES、192 ビットキーサイズ、CTR
  - **aes256-ctr**：AES、256 ビットキーサイズ、CTR
  - **aes128-gcm@openssh.com**：AES、128 ビットキーサイズ、Galois Counter モード [GCM]、OpenSSH
  - **aes256-gcm@openssh.com**：AES、256 ビットキーサイズ、GCM、OpenSSH
  - **chacha20-poly1305@openssh.com**：ChaCha20 対称暗号、Poly1305 暗号化メッセージ認証コード [MAC]、OpenSSH

通常のビルドにおけるアルゴリズムのデフォルトの文字列は次のとおりです。

```
blowfish-cbc,3des-cbc,aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,
chacha20-poly1305@openssh.com
```

信頼できるビルドにおけるアルゴリズムのデフォルトの文字列は次のとおりです。

```
aes256-ctr,aes192-ctr,aes128-ctr
```

ステップ 3 SSH 構成モードを終了します。

```
[local]host_name(config-sshd)# end
[local]host_name#
```

MAC アルゴリズムの設定

機能の概要と変更履歴

要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	• ASR 5500 • VPC-DI • VPC-SI
機能のデフォルト	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference* • *VPC-DI* システム管理ガイド • *VPC-SI System* アドミニストレーション ガイド

マニュアルの変更履歴

重要 リリース 21.2 および N5.1 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
最初の導入。	21.13

機能説明

MAC アルゴリズム設定機能を使用すると、内部 SSHD サーバーの MAC アルゴリズムの優先順位を設定または変更することができます。

この機能をサポートする、新しい CLI **MACs** CLI コマンドが SSH モード設定に導入されました。

MAC アルゴリズムの設定

ここでは、MAC アルゴリズムの設定方法を説明します。

MAC アルゴリズムの優先順位を指定するには、次の設定を使用します。

```
configure
   context context_name
      server sshd
         macs algorithms
         end

default macs
```

注：

- *alogorithms*：1 ～ 511 文字の英数字文字列を参照します。この文字列は、次のリストで示す優先順位（左から右）のコンマ区切りの変数（スペースなし）の 1 つの文字列として使用するアルゴリズムを指定します。
  - HMAC = ハッシュベースのメッセージ認証コード
  - SHA2 = セキュア ハッシュ アルゴリズム 2
  - SHA1 = セキュア ハッシュ アルゴリズム 1
  - ETM = Encrypt-Then-MAC
  - UMAC = ユニバーサルハッシュに基づくメッセージ認証コード
- 次に、通常のビルドのヘルプ文字列とアルゴリズムのリストを示します。

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1,umac-128-etm@openssh.com,umac-128@openssh.com,umac-64-etm@openssh.com,umac-64@openssh.com
  ```

- 次に、信頼できるビルドのヘルプ文字列とアルゴリズムのリストを示します。

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1
  ```

- デフォルト値の文字列は次のとおりです。

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1
  ```

MAC アルゴリズムの指定

MAC アルゴリズムの優先順位を設定するには、次の CLI コマンドを使用します。このコマンドは、SSH 構成モードで設定します。

```
configure
   context context_name
      server sshd
         macs algorithms
         end

default macs
```

注：

- *alogorithms*：1 ～ 511 文字の英数字文字列を参照します。この文字列は、次のリストで示す優先順位（左から右）のコンマ区切りの変数（スペースなし）の 1 つの文字列として使用するアルゴリズムを指定します。
  - HMAC = ハッシュベースのメッセージ認証コード
  - SHA2 = セキュア ハッシュ アルゴリズム 2
  - SHA1 = セキュア ハッシュ アルゴリズム 1
  - ETM = Encrypt-Then-MAC
  - UMAC = ユニバーサルハッシュに基づくメッセージ認証コード
- 次に、通常のビルドのヘルプ文字列とアルゴリズムのリストを示します。

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1,umac-128-etm@openssh.com,umac-128@openssh.com,umac-64-etm@openssh.com,umac-64@openssh.com
  ```

- 次に、信頼できるビルドのヘルプ文字列とアルゴリズムのリストを示します。

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1
  ```

- デフォルト値の文字列は次のとおりです。

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1
  ```

SSH キーの生成

ssh generate コマンドは、SSH サーバーによって使用される公開キーまたは秘密キーのペアを生成します。**ssh generate** CLI コマンド内に隠されている **v2-dsa** キーワード。SSH キーを生成するために使用できる唯一のキーワードは、**v2-rsa** です。

重要 生成されたキーペアは、コマンドが再度発行されるまで使用中のままになります。

手順

ステップ 1 コンテキスト構成モードを開始します。

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

ステップ 2 SSH キーペアを生成します。

```
[local]host_name(config-ctx)# ssh generate key type v2-rsa
[local]host_name(config-ctx)#
```

SSH キーペアの設定

ssh key コマンドは、システムで使用される公開キーと秘密キーのペアを設定します。**v2-dsa** キーワードは、**ssh key** コマンドでは隠されています。

手順

SSH キーペアのパラメータを指定します。

```
[local]host_name(config-ctx)# ssh key data length octets type v2-rsa
```

注：

- *data* は 1 ～ 1023 文字の英数字の文字列で表される暗号化キーです。
- **length** *octets* は 0 ～ 65535 の整数で表される、暗号化されたキーのオクテット単位の長さです。
- **type** はキータイプを指定します。**v2-rsa** はサポートされている唯一のタイプです。

重要 最大 200 の設定可能な承認済みの SSH キーが StarOS でサポートされています。

承認済み SSH ユーザーアクセス

ユーザーが、SSH 認証キーペアを持つ特定のホストから StarOS コンテキストにアクセスすることを許可する必要があります。

SSH ユーザーアクセスの認可

SSH 構成モードの **authorized-key** コマンドは、指定されたホストからのコンテキストへのユーザーアクセスを許可します。

手順

ステップ 1 SSH 構成モードに移動します。

```
[local]host_name(config-ctx)# server sshd
[local]host_name(config-sshd)#
```

ステップ 2 **authorized-key** コマンドを使用して管理ユーザーアクセスを指定します。

```
[local]host_name(config-sshd)# authorized-key username user_name host host_ip [ type { v2-dsa | v2-rsa } ]
```

注：

- **username** *user_name* は、sshd サーバーへのアクセスに許可されたキーを持つ既存の StarOS 管理者ユーザー名を指定します。*user_name* は、1 ～ 255 文字の英数字文字列で表されます。sshd キーをバイパス

しないようにするには、**nopassword** オプションを使用してコンテキスト構成モードの **administrator** コマンドを使用して、ユーザー名を事前に作成しておく必要があります。管理者の作成の詳細については、「システム設定」の章を参照してください。

- **host** *host_ip* は、このユーザー名の認証キーを持つ SSH ホストの IP アドレスを指定します。この IP アドレスは、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記である必要があります。
- **type** はキータイプを指定します。**v2-rsa** はサポートされている唯一のタイプです。

SSH ユーザーログインの制限事項

管理者は、StarOS CLI への SSH アクセスを、許可されたユーザーの「ホワイトリスト」に制限できます。サービスへのアクセスは、正当なニーズを持つユーザーにのみ制限される場合があります。明示的に許可されたユーザーのみが、SSH を介してホストに接続できます。ユーザー名には、必要に応じて特定の送信元 IP アドレスを含めることができます。

AllowUsers リストは、スペースで区切られたユーザー名パターンで構成されます。パターンで「USER」という形式を使用すると、そのユーザーに対してログインが制限されます。パターンが「USER@IP_ADDRESS'」形式の場合、ユーザーと IP アドレスは個別にチェックされ、指定した IP アドレスからのユーザーへのログインを制限します。

デフォルトでは、任意のユーザーによる無制限のアクセスを許可します。

許可済みユーザーリストの作成

allowusers add コマンドを使用すると、管理者は StarOS CLI にログインできるユーザーのリストを作成できます。

手順

ステップ 1 コンテキスト構成モードを開始します。

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

ステップ 2 SSH 構成モードに移動します。

```
[local]host_name(config-ctx)# server sshd
```

ステップ 3 SSH ユーザーリストを設定します。

```
[local]host_name(config-sshd)# allowusers add user_list
```

user_list は、スペースで区切られたユーザー名のパターンのリストを、1 ～ 999 文字の英数字の文字列として指定します。パターンで「USER」という形式を使用すると、そのユーザーに対してログインが制限されます。

パターンが「USER@IP_ADDRESS」形式の場合は、ユーザー名と IP アドレスが個別にチェックされ、その特定の IP アドレスからユーザーへのログインが制限されます。

パターンが「USER@<context>@IP_ADDRESS」形式の場合は、ユーザー名、StarOS コンテキスト、および IP アドレスが個別にチェックされ、その特定の IP アドレスから特定のコンテキストに関連付けられているユーザーへのログインを制限します。

user_list には次の制限が適用されます。

- この文字列の最大長は 3000 バイト（スペースを含む）です。
- スペースでカウントされる AllowUsers の最大数は256で、これは OpenSSH からの制限と一致します。

重要　上記の制限のいずれかを超えると、エラーメッセージが表示されます。このメッセージでは、正規表現のパターンを使用して文字列を短くするか、または **no allowusers add** や **default allowusers add** を使用してすべての allowusers を削除するか、または再設定するように求められます。

詳細については、『*Command Line Interface Reference*』の「*SSH Configuration Mode Commands*」の章を参照してください。

ステップ 4　SSH 構成モードを終了します。

```
[local]host_name(config-sshd)# end
[local]host_name#
```

SSH ユーザーログイン認証

StarOS は、次のシナリオの場合、許可済みキーとユーザーアカウントの組み合わせを使用して SSH によるユーザーログインの試行を認証します。

- ユーザーは、ローカルコンテキスト（VPN）インターフェイスを介してローカルコンテキストのユーザー名と、ローカルコンテキストで設定されている許可済みのキーを使用してログインしようとします。
- ユーザーは、ローカル以外のコンテキストインターフェイスを介してローカル以外のコンテキストのユーザー名と、ローカル以外のコンテキストで設定されている許可済みのキーを使用してログインしようとします。
- ユーザーは、ローカル以外のコンテキストインターフェイスを介してローカルコンテキストのユーザー名と、ローカルコンテキストで設定されている許可済みのキーを使用してログインしようとします。
- ユーザーは、ローカル コンテキスト インターフェイスを介してローカル以外のコンテキストのユーザー名と、ローカル以外のコンテキストで設定されている許可済みのキーを使用してログインしようとします。

現在のシステム設定に基づいて認証が失敗すると、ログインが阻止され、エラーメッセージが生成されます。

StarOS では、ユーザー ID が異なるユーザーが同じ公開 SSH キーを使用して、許可されていないコンテキストへログインすることは許可されていません。ユーザーの認証では、許可済みキーとユーザーアカウントの組み合わせが考慮されます。

重要 StarOS リリース 21.0 以降では、ユーザーがローカル以外のコンテキストからログインした場合、そのユーザーは /flash ディレクトリにアクセスできません。

セキュアなセッションログアウト

StarOS が SSH クライアントから切断されると、デフォルトの動作によって CLI または SFTP セッションは約 45 秒（デフォルトのパラメータを使用）で終了します。SSH 構成モードの CLI コマンドを使用すると、このデフォルトの SSHD 切断動作を無効にしたり、変更したりできます。

重要 セキュリティを強化するため、シスコでは、少なくとも lient-alive-countmax を 2、client-alive-interval を 5 にすることを推奨します。セッションのログアウト値が小さいと、ssh セッションのログアウトが不定期にログアウトする可能性があります。セキュリティとユーザーの使いやすさとのバランスが取れるように値を調整します。

client-active-countmax コマンドは、sshd なしで送信される client-alive メッセージの数を、SSH クライアントからのメッセージを受信しないように設定します（デフォルトは 3）。client-alive メッセージの送信中にこのしきい値に達すると、sshd は SSH クライアントを切断してセッションを終了します。

client-alive-interval コマンドは、タイムアウト間隔を秒単位で設定します（デフォルトは 15）。その後、SSH クライアントからデータを受信しなかった場合、sshd は暗号化されたチャネルを介してメッセージを送信し、クライアントからの応答を要求します。メッセージが送信される回数は、client-alive-countmax パラメータによって決定されます。sshd が SSH クライアントの切断を解除するまでのおおよその時間は、client-alive-countmax X client-alive-interval となります。

クライアントまたはサーバーがいつ接続が非アクティブになったかを認識しているかどうかに依存している場合、client-alive メカニズムは重要です。

重要 client-alive メッセージは暗号化チャネルを介して送信されるため、スプーフィングできません。

重要 これらのパラメータは、SSH プロトコルバージョン 2 のみに適用されます。

デフォルトの sshd セキュア セッション ログアウト パラメータの変更

次のコマンドシーケンスは、クライアントの ClientAliveCountmax（デフォルトは 3）および ClientAliveInterval（デフォルトは 15 秒）のパラメータのデフォルト設定を変更します。

手順

ステップ 1 コンテキスト構成モードを開始します。

```
[local]host_name# configure
```

ステップ 2 SSH 構成モードに移動します。

```
[local]host_name(config)# context context_name
```

ステップ 3 ClientAliveCountmax パラメータを 2 に設定します。

```
[local]host_name(config-sshd)# client-alive-countmax 2
```

ステップ 4 ClientAliveInterval パラメータを 5 秒に設定します。

```
[local]host_name(config-sshd)# client-alive-interval 5
```

ステップ 5 SSH 構成モードを終了します。

```
[local]host_name(config-sshd)# end
[local]host_name#
```

SSHD キーボードインタラクティブ認証

SSHD 設定のチャレンジレスポンス認証オプションは、キーボードインタラクティブ認証方式を有効にするために使用されます。この認証方式は、特定のケースで役に立ちます。たとえば、TACACS サーバがシステムにログインするユーザーとの対話を必要とする場合などです。

キーボードインタラクティブ認証方式の有効化

手順

ステップ 1 コンテキスト構成モードを開始します。

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

ステップ 2 SSH 構成モードに移動します。

```
[local]host_name(config-ctx)# server sshd
```

ステップ 3 チャレンジレスポンス認証を設定します。

```
[local]host_name(config-sshd)# challenge-response-authentication
```

SSHD チャレンジレスポンス認証を指定し、レガシーの PGW、SGW、または SAEGW に対してのみ有効にします。詳細については、『*Command Line Interface Reference*』の「*SSHD Configuration Mode Commands* 」の章を参照してください。

ステップ 4 SSH 構成モードを終了します。

```
[local]host_name(config-ctx)# end
[local]host_name#
```

不具合

1. チャレンジレスポンス認証オプションは、リリース 21.28 以降でのみサポートされます。
2. チャレンジレスポンス認証の有効化は、特定の特殊なケースでのみ推奨されます。たとえば、TACACS サーバがユーザーに特定のプロンプトを表示することを選択した場合などです。特別な理由がない限り、チャレンジレスポンス認証は有効にしないでください。このオプションを有効にする前に、シスコの担当者にお問い合わせください。
3. 明示的に制限されていなくても、レガシーの PGW、SGW、SAEGW 以外の製品についてはチャレンジレスポンス認証を有効にしないことを強くお勧めします。
4. キーボードインタラクティブ認証方式を使用するには、SSH ログインに使用される IP アドレスを所有するコンテキストで、チャレンジレスポンス認証を有効にする必要があります。
5. チャレンジレスポンス認証は、コンソールを介した SSH ログインには影響しません。
6. チャレンジレスポンス認証は SSHD オプションであり、Telnet や FTP のログインには影響しません。
7. ユーザー応答のサイズは 128 バイト未満である必要があります。

重要 TACACS と組み合わせて使用されるチャレンジレスポンス認証も想定しており、これも特殊なケースになります。TACACS サーバは、[AUTHEN-REPLY Server Message] フィールドで最大 511 文字を送信でき、その文字はログインしようとしているエンドユーザーに渡されます。[Server Message] フィールドの長さが 512 バイト以上の場合、[ERROR: Enter any key to continue.] というエラーメッセージがユーザーに表示され、TACACS 認証は想定どおりに失敗します。

8. チャレンジレスポンス認証が有効になっている場合、ユーザーにはプロンプトへの応答のために 60 秒が与えられます。

外部サーバーへの SSH クライアントログイン

StarOS は、StarOS ゲートウェイから外部サーバーへの SSH/SFTP アクセスの公開キーの認証をサポートしています。この機能を設定するには、SSH クライアントキーのペアを生成し、クライアント公開キーを外部サーバーにプッシュします。

（注） デフォルトでは、StarOS は外部サーバーへの username-password の認証のみをサポートしています。

SSH クライアント暗号の設定

SSH クライアント構成モードの **cipher** CLI コマンドは、外部サーバーにログインするときに暗号優先順位リストを設定します。

手順

ステップ 1 SSH クライアント構成モードを開始します。

```
[local]host_name(config)# client ssh
```

ステップ 2 必要な暗号化アルゴリズムを指定します。

```
[local]host_name(config-ssh)# ciphers algorithms
```

注：

- アルゴリズムは 1 ～ 511 文字の英数字の文字列で、次に示すように、優先順位（左から右）でカンマ区切りの変数（スペースなし）の単一の文字列として使用するアルゴリズムを指定します。
  - **blowfish-cbc**：対称キーブロック暗号、暗号ブロック連鎖（CBC）
  - **3des-cbc**：トリプルデータ暗号化規格、CBC
  - **aes128-cbc**：Advanced Encryption Standard（AES; 高度暗号化規格）、128 ビットキーサイズ、CBC
  - **aes128-ctr**：AES、128 ビットキーサイズ、カウンタモード暗号化（CTR）
  - **aes192-ctr**：AES、192 ビットキーサイズ、CTR
  - **aes256-ctr**：AES、256 ビットキーサイズ、CTR
  - **aes128-gcm@openssh.com**：AES、128 ビットキーサイズ、Galois Counter モード [GCM]、OpenSSH
  - **aes256-gcm@openssh.com**：AES、256 ビットキーサイズ、GCM、OpenSSH
  - **chacha20-poly1305@openssh.com**：ChaCha20 対称暗号、Poly1305 暗号化メッセージ認証コード [MAC]、OpenSSH

通常のビルドにおけるアルゴリズムのデフォルトの文字列は次のとおりです。

```
aes256-ctr,aes192-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,
blowfish-cbc,3des-cbc,aes128-cbc
```

信頼できるビルドにおけるアルゴリズムのデフォルトの文字列は次のとおりです。

```
aes256-ctr,aes192-ctr,aes128-ctr
```

ステップ 3 SSH クライアント構成モードを終了します。

```
[local]host_name(config-ssh)# end
[local]host_name#
```

優先認証方式の設定

SSH クライアント構成モードの **preferredauthentications** CLI コマンドは、適切な認証方式を設定します。

手順

ステップ 1 SSH クライアント構成モードを開始します。

```
[local]host_name(config)# client ssh
```

ステップ 2 優先認証方式の指定

```
[local]host_name(config-ssh)# preferredauthentications methods
```

注：

- 方式：次に示すように、優先順位順（左から右）に、カンマ区切りの変数（スペースなし）の単一の文字列として使用される認証方式を指定します。
  - **publickey** ：SSH v2-RSA プロトコルを使用した認証
  - **keyboard-interactive** ：任意の数の情報を要求します。各情報について、サーバーはプロンプトのラベルを送信します。
  - **password** ：単一のパスワードの単純な要求
- **デフォルト**：方式の値を [publickey,password] にリセットします。

ステップ 3 SSH クライアント構成モードを終了します。

```
[local]host_name(config-ssh)# exit
[local]host_name(config)#
```

SSH クライアントキーペアの生成

SSH クライアント構成モードでコマンドを使用し、秘密キーを指定して、SSH クライアントキーペアを生成します。

手順

ステップ 1 SSH クライアント構成モードを開始します。

```
[local]host_name(config)# client ssh
[local]host_name(config-ssh)#
```

ステップ 2 SSH クライアントキーのペアを生成します。

```
[local]host_name(config-ssh)# ssh generate key [ type v2-rsa ] [ key-size ]
[local]host_name(config-ssh)#
```

type v2-rsa は SSH クライアントキーのタイプを指定します。サポートされている SSH クライアントキーのタイプは、**v2-rsa** のみです。

key-size は SSH クライアントのキーサイズを指定します。サポートされているキーサイズは、2048、3072、4096、5120、6144、7168、および 9216 です。

ステップ 3 SSH クライアントキーが生成されていることを確認します。

```
[local]host_name(config-ssh)# do show ssh client key
```

ステップ 4 SSH クライアント構成モードを終了します。

```
[local]host_name(config-ssh)# exit
[local]host_name(config)#
```

外部サーバーへの SSH クライアント公開キーのプッシュ

このサーバーへの SSH/SFTP アクセスをサポートするには、SSH クライアント公開キーを外部サーバーにプッシュする必要があります。

手順

ステップ 1 Exec モードで、**push ssh-key** コマンドを実行します。

```
[local]host_name# push ssh-key { host_name | host_ip_address } user username [ context context_name ]
[local]host_name#
```

host_name は、DNS ルックアップを介して解決される必要がある論理ホスト名を使用してリモートサーバーを指定します。これは、1 ～ 127 文字の英数字文字列で表されます。

host_ip_address は、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記で表されます。

user *username* は、外部サーバーで有効なユーザー名を 1 ～ 79 文字の英数字の文字列として指定します。

context *context_name*は、有効なコンテキスト名を指定します。コンテキスト名はオプションです。指定されていない場合は、現在のコンテキストが処理に使用されます。

ステップ 2 他の外部サーバーでの SSH/SFTP アクセスをサポートするには、ステップ 1 を繰り返します。

ステップ 3 外部サーバーへの SSH クライアントのログインをテストします。

```
local]host_name# ssh { hostname | ip_address } user username port port_number
```

NETCONF の有効化

SSH キーは、NETCONF プロトコルと ConfD エンジンが Cisco Network Service Orchestrator（NSO）をサポートするために有効になる前に必要になります。

NETCONF を有効にする方法の詳細については、このガイドの付録の「*NETCONF* と *ConfD*」を参照してください。

2 番目の IP アドレスを使用した管理インターフェイスの設定

必要に応じて、MIO/UMIO の管理インターフェイスで 2 番目の IP アドレスを設定できます。

手順

ステップ 1 プロンプトで次のコマンドを入力して、構成モードを開始します。

```
[local]host_name#  configure
[local]host_name(config)#
```

ステップ 2 コンテキスト構成モードを開始するには、次のように入力します。

```
[local]host_name(config)# context local
[local]host-name(config-ctx)#
```

ステップ 3 次のコマンドを入力して、インターフェイスのスロット番号とポート番号を入力します。

```
[local]host_name(config-ctx)# 5/1
[local]host_name(config-if-eth)#
```

ステップ 4 次のコマンドを入力して、セカンダリ IP アドレスとサブネットマスクを入力します。

```
[local]host_name(config-if-eth)# { ip | ipv } address ipaddress subnet_mask secondary
```

ステップ 5 次のコマンドを入力して、構成モードを終了します。

```
[local]host_name(config-if-eth)# end
```

ステップ 6 次のコマンドを入力して、インターフェイスの IP アドレスを確認します。

```
[local]host_name# show config context local
```

CLI 出力は次の例のようになります。

```
config
   context local
      interface interface_name
         ip address ipaddress subnetmask
         ip address ipaddress subnetmask secondary
         #exit
```

ステップ 7 「設定の確認と保存」の説明に従って、設定を保存します。

2 番目の IP アドレスを使用した管理インターフェイスの設定

必要に応じて、vNIC 管理インターフェイスに 2 番目の IP アドレスを設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	プロンプトで次のコマンドを入力して、構成モードを開始します。	`[local]host_name` **configure** `[local]host_name(config)`
ステップ 2	コンテキスト構成モードを開始するには、次のように入力します。	`[local]host_name(config)` **context local** `[local]host-name(config-ctx)`
ステップ 3	次のコマンドを使用して、インターフェイスのスロット番号とポート番号を入力します。	`[local]host_name(config-ctx) 1/1` `[local]host_name(config-if-eth)`
ステップ 4	次のコマンドを入力して、セカンダリ IP アドレスとサブネットマスクを入力します。	`[local]host_name(config-if-eth)` **{ ip | ipv } address** *ipaddress subnet_mask* secondary
ステップ 5	次のコマンドを入力して、構成モードを終了します。	`[local]host_name(config-if-eth)` **end**

ステップ 6	次のコマンドを入力して、インターフェイスの IP アドレスを確認します。	`[local]host_name` **show config context local**
		CLI 出力は次の例のようになります。
		`config`
		`context local`
		`interface interface_name`
		`ip address ipaddress subnetmask`
		`ip address ipaddress subnetmask secondary`
		`exit`

ステップ 7	インターフェイスとポートの設定の確認と保存に進みます。	

Open SSH から Cisco SSH へのアップグレードと移行

機能の概要と変更履歴

要約データ

該当製品または機能エリア	すべて
該当プラットフォーム	• ASR 5500 • VPC-DI • VPC-SI
機能のデフォルト	有効、常時オン
このリリースでの関連する変更点	N/A
関連資料	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference* • *VPC-DI* システム管理ガイド • *VPC-SI System* アドミニストレーション ガイド

マニュアルの変更履歴

重要 リリース 21.2 および N5.1 よりも前に導入された機能の改訂履歴の詳細は示していません。

改訂の詳細	リリース
このリリースでは、暗号と MAC のアルゴリズム値は、OpenSSH から CiscoSSH へのアップグレードと移行に基づいて変更されています。	21.16
最初の導入。	21.2 よりも前

変更された機能

Cisco ASR 5500 および VPC 製品のセキュリティ対策として、暗号および MAC アルゴリズム値は、Cisco SSH バージョンへの Open SSH のアップグレードと移行をサポートするように変更されています。

以前の動作：21.16よりも前のリリースでは、**cipher**コマンドと**macs**コマンドの**default**アルゴリズム値は次のようになっていました。

- **暗号化方式**

  21.15（通常ビルドのみ）

  通常のビルドのアルゴリズムの値を次のようにリセットします。

  ```
  blowfish-cbc,3des-cbc,aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-poly1305@openssh.com
  ```

- **MAC**

  21.15（信頼できるビルドのみ）

  信頼できるビルドのアルゴリズムの値を次のようにリセットします。

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1
  ```

- **KEX アルゴリズム**

  21.15

  通常のビルドと信頼できるビルドで使用可能なアルゴリズム：

  ```
  diffie-hellman-group1-sha1,diffie-hellman-group14-sha1
  ```

新しい動作：このリリースでは、**default** コマンドと **cipher** コマンドの**macs**アルゴリズム値は次のとおりです。

- **暗号化方式**

  リリース 21.16 以降：**Post OpenSSH から CiscoSSH へのアップグレードと移行**

  通常のビルドのデフォルトのアルゴリズムは次のとおりです。

  ```
  aes256-ctr,aes192-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com
  ```

  通常のビルドで使用可能なアルゴリズムは次のとおりです。

  ```
  aes256-ctr,aes192-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc
  ```

  信頼できるビルドでデフォルトのアルゴリズムと使用可能なアルゴリズム：

  ```
  aes256-ctr,aes192-ctr,aes128-ctr
  ```

（注） 信頼できるビルドのデフォルトの暗号と設定可能な暗号に変更はありません。

- **MAC**

  リリース 21.16 以降：**Post OpenSSH から CiscoSSH へのアップグレードと移行**

  通常のビルドでデフォルトのアルゴリズムと使用可能なアルゴリズム：

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1
  ```

信頼できるビルドでデフォルトのアルゴリズム：

```
hmac-sha2-512,hmac-sha2-256,hmac-sha1
```

信頼できるビルドで使用可能なアルゴリズム：

```
hmac-sha2-512,hmac-sha2-256,hmac-sha1
```

（注） hmac-sha2-512-etm@openssh.com、hmac-sha2-256-etm@openssh.com、hmac-sha1-etm@openssh.com は信頼できるビルドから削除されます。

- **KEX アルゴリズム**

  リリース **21.16** 以降：**Post OpenSSH から CiscoSSH へのアップグレードと移行**

  通常のビルドと信頼できるビルドで使用可能なアルゴリズム：

```
diffie-hellman-group14-sha1
```

（注） KEX アルゴリズムは、StarOS では設定できません。したがって、CLI の変更はありません。

VM ハードウェアの検証

リソース割り当ての問題を回避するには、システム内で使用されるすべての VM が同じサイズの CPU と同じサイズのメモリを持つことが重要です。すべてのインターフェイスでパフォーマンスのバランスを取るために、サービスポートと DI ポートが同じスループット能力を備えていることを確認してください。

すべてのカードまたは特定のカードのハードウェア設定を確認するには、**show cloud hardware**[*card_number*} コマンドを使用します。次に、カード 1（CF）でのこのコマンドの出力例を示します。

```
[local]s1# show cloud hardware 1

Card 1:
   CPU Nodes               : 1
   CPU Cores/Threads       : 8
   Memory                  : 16384M (qvpc-di-medium)
   Hugepage size           : 2048kB
   cpeth0                  :
     Driver                : virtio_net
   loeth0                  :
     Driver                : virtio_net
```

次に、カード 3（SF）でのこのコマンドの出力例を示します。

```
[local]s1# show cloud hardware 1
```

```
Card 3:
   CPU Nodes               : 1
   CPU Cores/Threads       : 8
   Memory                  : 16384M (qvpc-di-medium)
   Hugepage size           : 2048kB
   cpeth0                  :
     Driver                : vmxnet3
   port3_10                :
     Driver                : vmxnet3
   port3_11                :
     Driver                : vmxnet3
```

基本となる VM ハードウェアの最適な設定を表示するには、**show hardware optimum** を使用します。現在の VM 設定を最適な設定と比較するには、**show cloud hardware test** コマンドを使用します。最適に設定されていないパラメータは、次の出力例に示すように、アスタリスク付きでフラグが立てられます。この例では、CPU コア/スレッドおよびメモリが最適に設定されていません。

```
[local]s1# show cloud hardware test 1

Card 1:
   CPU Nodes               : 1
*  CPU Cores/Threads       : 8                              Optimum value is 4
*  Memory                  : 8192M (qvpc-di-medium)         Optimum value is 16384
   Hugepage size           : 2048kB
   cpeth0                  :
     Driver                : virtio_net
   loeth0                  :
     Driver                : virtio_net
```

設定ディスクまたはローカルフラッシュ上の設定ファイルを表示するには、**show cloud configuration** *card_number* コマンドを使用します。フラッシュメモリ上のロケーション パラメータ ファイルは、インストール時に定義されます。また、ディスク構成は通常、オーケストレーションによって作成され、カードに接続されます。次に、カード 1 でのこのコマンドの出力例を示します。

```
[local]s1# show cloud configuration 1

Card 1:
  Config Disk Params:
-------------------------
    No config disk available

  Local Params:
-------------------------
CARDSLOT=1
CARDTYPE=0x40010100
CPUID=0
```

すべてのカードまたは特定のカードの IFTASK 設定を表示するには、**show cloud hardware iftask** コマンドを使用します。デフォルトでは、コアは PMD と VNPU の両方に使用されるように設定されています。次に、カード 4 でのこのコマンドの出力例を示します。

```
[local]mySystem# show cloud hardware iftask 4
Card 4:
  Total number of cores on VM:      24
  Number of cores for PMD only:     0
```

```
Number of cores for VNPU only:    0
Number of cores for PMD and VNPU: 3
Number of cores for MCDMA:        4
Hugepage size:           2048 kB
Total hugepages:     16480256 kB
NPUSHM hugepages:           0 kB
CPU flags: avx sse sse2 ssse3 sse4_1 sse4_2
Poll CPU's: 1 2 3 4 5 6 7
KNI reschedule interval: 5 us
```

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。