

Oracle® Solaris カーネルゾーンの作成と使用

ORACLE®

Part No: E62814
2016年11月

Part No: E62814

Copyright © 2014, 2016, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、Oracle Corporationおよびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはオラクル およびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

目次

このドキュメントの使用方法	9
1 Oracle Solaris カーネルゾーンの計画と構成	11
Oracle Solaris カーネルゾーンについて	11
カーネルゾーンと一般的なゾーンの概念	12
Oracle Solaris カーネルゾーンのハードウェアおよびソフトウェア要件	13
カーネルゾーンホストでのハードウェアおよびソフトウェアサポート の確認	15
カーネルゾーンのメモリーを予約するための ZFS ARC のチューニン グ	16
Oracle Solaris カーネルゾーンの構成	17
▼ カーネルゾーンを構成する方法	17
カーネルゾーンリソースの構成およびカスタマイズ	18
カーネルゾーンの CPU の管理	19
カーネルゾーンのメモリーの管理	21
カーネルゾーンのストレージデバイスとブート順序の管理	25
カーネルゾーンのネットワークデバイスと構成の管理	26
カーネルゾーンでのシングルルート I/O NIC 仮想化の管理	28
▼ 単一の anet を使用してカーネルゾーンで SR-IOV NIC 仮想機能を 有効にする方法	28
Oracle Solaris カーネルゾーンでの仮想機能とシャドウ VNIC の使用	30
カーネルゾーンでの仮想 LAN の構成	31
カーネルゾーンでの動的な MAC アドレスと VLAN ID の使用	33
▼ 動的な MAC アドレスおよび VLAN ID をカーネルゾーンの anet 構 成に使用する方法	33
IPoIB とカーネルゾーンの操作	34
suspend リソースの構成	34
ベリファイドブートを使用した Oracle Solaris カーネルゾーンのセキュリ ティー保護	36
verified-boot リソースプロパティーおよび elfsign 検証につい て	36

カーネルゾーンでの Silicon Secured Memory の有効化	38
2 Oracle Solaris カーネルゾーンのインストール、停止処理、およびクローニング	41
カーネルゾーンのインストール	41
直接インストールを使用したカーネルゾーンのインストール	42
カーネルゾーンインストールでの AI マニフェストと sysconfig プロファイルの使用	44
インストールイメージからのカーネルゾーンのインストール	48
カーネルゾーンのアンインストール	49
カーネルゾーンの停止処理、リブート、および停止	49
カーネルゾーンのクローニング	50
3 Oracle Solaris カーネルゾーンの移行	53
カーネルゾーンの移行の要件	53
コールド移行を使用したカーネルゾーンの移行	53
ウォーム移行を使用したカーネルゾーンの移行	54
▼ ウォーム移行を使用してカーネルゾーンを移行する方法	54
ライブ移行を使用したカーネルゾーンの移行	55
ライブ移行の要件	56
root 以外のユーザーによるカーネルゾーンのライブ移行操作の実行の許可	57
zoneadm migrate コマンドについて	57
▼ ライブ移行を使用してカーネルゾーンを移行する方法	58
セキュアなライブ移行について	62
暗号化鍵とカーネルゾーンの移行	63
SPARC カーネルゾーンのウォーム移行およびライブ移行用の CPU 移行クラス の指定	63
4 Oracle Solaris カーネルゾーンの管理	65
カーネルゾーン環境での作業	65
ゾーン内でのプロセス ID の可視性	65
カーネルゾーンの重複プロセス ID	66
カーネルゾーンのゾーンパス	66
カーネルゾーンのリソース管理機能	66
カーネルゾーンと不変ゾーンの操作	66
カーネルゾーンでのリムーバブルデバイスの管理	67
▼ 仮想 CD-ROM デバイスをカーネルゾーンに追加する方法	67

カーネルゾーンの補助状態の操作	69
入れ子にされたゾーンの管理	70
▼ 複数の MAC アドレスをカーネルゾーンに追加する方法	71
入れ子にされたゾーンおよび新しい非大域ゾーンの構成	71
カーネルゾーンのホストデータおよびホスト ID	72
カーネルゾーンのブートローダーの操作	73
▼ カーネルゾーンで代替ブート環境を指定する方法	73
カーネルゾーンのライブゾーン再構成	75
NFS ストレージ URI とカーネルゾーン	75
NFS ストレージ URI の構文と使用法	76
カーネルゾーン内のコアファイル	76
索引	77

このドキュメントの使用方法

- **概要** – Oracle Solaris ゾーンを計画、構成、インストール、および管理する方法について説明します
- **対象読者** – 技術者、システム管理者、および認定サービスプロバイダ
- **必要な知識** – Oracle Solaris 環境の管理経験。仮想化環境の経験も推奨されます。

製品ドキュメントライブラリ

この製品および関連製品のドキュメントとリソースは <http://www.oracle.com/pls/topic/lookup?ctx=E62101-01> で入手可能です。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

◆◆◆ 第 1 章

Oracle Solaris カーネルゾーンの計画と構成

この章では、Oracle Solaris カーネルゾーン (別名 `solaris-kz` ブランドゾーン) を計画および構成する方法について説明します。ここでは、ハードウェアサポートの確認、ゾーンの構成、およびカーネルゾーンに固有のゾーンリソースの適用の手順を説明します。

この章では、次の計画および構成について説明します。

Oracle Solaris カーネルゾーンについて

Oracle Solaris カーネルゾーン (別名 `solaris-kz` ブランドゾーン) は、ブランドゾーンフレームワークを使用して、大域ゾーンからの個別のカーネルとオペレーティングシステム (OS) のインストールでゾーンを実行します。個別のカーネルおよび OS インストールにより、オペレーティングシステムのインスタンスとアプリケーションの独立性が高まり、セキュリティが強化されます。

カーネルゾーンの管理および構造的な内容は、大域ゾーンの内容にはまったく依存していません。たとえば、カーネルゾーンでは、大域ゾーンやカーネルゾーンホストとシステムパッケージングを共有しません。カーネルゾーンホストでのパッケージ更新は、リンクイメージにはならず、カーネルゾーンにも影響しません。同様に、`pkg update` などのパッケージングコマンドは、カーネルゾーン内側から完全に機能します。パッケージングコマンドの詳細は、[『Oracle Solaris 11.3 での Image Packaging System を使用したソフトウェアのパッケージ化と配布』の第 3 章、「ソフトウェアパッケージのインストール、削除、および更新」](#)を参照してください。

システムプロセスは、カーネルゾーンの個別のプロセス ID テーブルで処理され、大域ゾーンとは共有されません。カーネルゾーンのリソース管理も異なります。`max-processes` などのリソース制御は、カーネルゾーンを構成する場合に使用できません。

`zoneadm rename` コマンドは、インストール済み状態のカーネルゾーンではサポートされません。`zonectfg` コマンドを使用することによってだけ、カーネルゾーンの名前を変更できます。カーネルゾーンは、構成済みまたは利用不可能な状態になっている必要があります。

大域ゾーンでカーネルゾーンを管理するには、既存の `zlogin`、`zonecfg`、および `zoneadm` コマンドを使用します。

ブランドゾーンフレームワークの詳細は、[brands\(5\)](#) のマニュアルページを参照してください。

カーネルゾーンの概念の概要については、『[Oracle Solaris ゾーン](#)の紹介』の第1章、『[Oracle Solaris ゾーン](#)の概要』を参照してください。



注意 - Oracle Solaris x86 システムでは、Oracle VM VirtualBox と Oracle Solaris カーネルゾーンを同時に実行しないでください。

カーネルゾーンと一般的なゾーンの概念

このマニュアルでは、次のリソース管理とゾーンの概念に精通していることを前提とします。

- 利用可能なシステムリソースをアプリケーションでどのように使用するかを決定するリソース制御
- ゾーン構成、インストール、および管理に使用されるコマンド。主に `zonecfg`、`zoneadm`、および `zlogin`
- `zonecfg` リソースおよびプロパティタイプ
- 大域ゾーンおよび非大域ゾーン
- 完全ルート非大域ゾーンモデル
- 承認は、`zonecfg` ユーティリティを使用して付与します。
- 大域管理者およびゾーン管理者
- ゾーンの状態モデル
- ゾーン隔離の特性
- ネットワークの概念と構成
- ゾーン排他的 IP タイプ

これらの概念の詳細は、『[Oracle Solaris ゾーン](#)の紹介』および『[Oracle Solaris ゾーン](#)の作成と使用』を参照してください。

Oracle Solaris カーネルゾーンのハードウェアおよびソフトウェア要件

ライブ移行など、カーネルゾーンで最新の機能を使用するには、ホストオペレーティングシステムで少なくとも Oracle Solaris 11.3 を実行している必要があります。

物理マシンが次の要件を満たしている必要があります。

SPARC ベースのシステム:

- システムファームウェア 8.8 以上がインストールされている SPARC T4 シリーズサーバー。
- システムファームウェア 9.5 以上がインストールされている SPARC T5、SPARC M5、または SPARC M6 シリーズサーバー。
- SPARC T7 または SPARC M7 シリーズサーバー。すべてのファームウェアバージョンがサポートされています。
- カーネルゾーンを実行する場合、XCP ファームウェア 2230 以降および Oracle Solaris 11.3 以降を使用する Fujitsu M10 または SPARC M10 サーバー。

Fujitsu M10 システムでカーネルゾーンのライブ移行機能を使用するには、*Fujitsu M10* システムのプロダクトノートファームウェア要件に従います。

最新のシステムファームウェアのダウンロードについては、[Oracle システムのファームウェアのダウンロードとリリース履歴](#)に関する Web ページを参照してください。

x86 ベースのシステム:

- Intel ベースシステムには Nehalem またはそれ以降のプロセッサが必要です
- AMD ベースシステムには Barcelona またはそれ以降のプロセッサが必要です
- BIOS では、次の機能が有効になっている必要があります。
 - CPU 仮想化 (VT-x など)
 - EPT、NPT、または RVI (Rapid Virtualization Indexing) と呼ばれる Extended/Nested Page Table のサポート
 - NX、XD、No-Execute Memory Protection、No Execute Mode Mem Protection、Execute Disable、または Execute Bit サポートとも呼ばれる、No-eXecute のサポート

SPARC および x86 ベースのシステムでは次のものが必要です。

- 8G バイト以上の物理 RAM
- カーネルゾーンのブランドソフトウェアパッケージ brand/brand-solaris-kz。ソフトウェアパッケージの取得およびインストールについては、『[Oracle Solaris 11.3 での Image Packaging System を使用したソフトウェアのパッケージ化と配布](#)』

の第3章、「ソフトウェアパッケージのインストール、削除、および更新」を参照してください。

- リモート管理デーモン (RAD) を使用するには、rad-zonemgr パッケージがシステムにインストールされている必要があります。システム間で発生するゾーン移行などの操作については、rad-zonemgr パッケージがターゲットシステムとソースシステムの両方にインストールされている必要があります。RAD モジュールのインストール後、svcadm restart rad コマンドを使用して RAD SMF サービスを手動で再起動する必要があります。
- メモリーエラーを回避するには、カーネルゾーンホスト上で ZFS Adaptive Replacement Cache (ARC) のパラメータを調整する必要があります。16 ページの「カーネルゾーンのメモリーを予約するための ZFS ARC のチューニング」を参照してください。

カーネルゾーンは、大域ゾーンのパブリッシャーとデフォルトの AI マニフェスト、カスタム AI マニフェスト、Oracle Solaris インストールメディアの ISO イメージ、または統合アーカイブを使用してインストールできます。

-a、-b、または -m オプションが指定されないかぎり、インストールの実行にはデフォルトの AI マニフェスト /usr/share/auto_install/manifest/default.xml と、大域ゾーンの pkg パブリッシャーが使用されます。サポートされているインストーラは、テキストインストーラおよび Automated Installer です。これにより、サポートされている任意の Oracle Solaris バージョンをインストールできます。Oracle Solaris 11.2 はカーネルゾーンでサポートされる最初のバージョンの Oracle Solaris です。

Oracle Solaris ゾーンは、Oracle VM Server for SPARC (以前は「Sun 論理ドメイン」と呼ばれていました) 上のゲストで実行できます。各 Oracle VM Server for SPARC ドメインには、実行できるカーネルゾーンの数に個別の制限があります。制限は、SPARC T4 または SPARC T5 システムでは 768、SPARC M5 または SPARC M6 システムでは 512、Fujitsu M10 システムでは 256 です。

カーネルゾーンは、Oracle VM Server for x86 ゲスト内、または Oracle VM VirtualBox 上で実行できません。

注記 - SPARC ベースシステムでは、Oracle VM Server for SPARC ドメイン内で実行中のカーネルゾーンによってゲストドメインの Oracle VM Server for SPARC ライブ移行がブロックされます。詳細は、Oracle Solaris 11.3 のリリースノートを参照してください。

SPARC ベースシステムでのカーネルゾーンのライブ移行には、追加のソフトウェアおよびファームウェア要件があります。56 ページの「ライブ移行の要件」を参照してください。

注記 - システム上でさまざまなゾーンブランドを実行できますが、カーネルゾーンを実行する場合は、ゾーンの実行用にカーネルゾーンホストを予約し、大域ゾーンでアプリケーションを実行しないようにしてください。

カーネルゾーンホストでのハードウェアおよびソフトウェアサポートの確認

カーネルゾーンを計画して配備する前に、13 ページの「Oracle Solaris カーネルゾーンのハードウェアおよびソフトウェア要件」の説明に従って、カーネルゾーンホストにハードウェアおよびソフトウェア要件があることを確認する必要があります。virtinfo コマンドを使用すると、カーネルゾーンホストのハードウェア要件、ファームウェアまたは BIOS 要件、およびカーネルゾーンのブランドパッケージのソフトウェア要件を確認できます。

▼ システムでサポートできるカーネルゾーンを確認する方法

1. カーネルゾーンホストで、管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. Oracle Solaris オペレーティングシステムのバージョンが 11.2 以上であることを確認します。

```
# uname -a
```

たとえば、システム global で:

```
global# uname -a
SunOS global 5.11 11.2 sun4v sparc sun4v
```

3. カーネルゾーンのブランドパッケージ brand/brand-solaris-kz のインストールを確認します。

```
# pkg list brand/brand-solaris-kz
```

次の例は、カーネルゾーンブランドパッケージが system global にインストールされていることを示しています。

```
global# pkg list brand/brand-solaris-kz
NAME (PUBLISHER)                                VERSION                                IFO
system/zones/brand/brand-solaris-kz           0.5.11-0.175.2.0.0.36.22321         i--
```

4. virtinfo コマンドを実行します。

```
# virtinfo
```

次の例の出力は、カーネルゾーンが論理ドメインであるシステム global でサポートされていることを示しています。

```
global# virtinfo
NAME          CLASS
logical-domain current
non-global-zone supported
```

kernel-zone supported

参照 詳細は、[virtinfo\(1M\)](#) のマニュアルページを参照してください。

カーネルゾーンのメモリーを予約するための ZFS ARC のチューニング

カーネルゾーンの効率的なパフォーマンスを確保するため、カーネルゾーンをホストしているシステムで `user_reserve_hint_pct` チューニング可能パラメータを設定する必要があります。パラメータは、システムにアプリケーションのメモリー使用に関するヒントを提供し、アプリケーションに使用できるメモリーがより多く残るように、ZFS Adaptive Replacement Cache (ARC) の増加を制限するために使用します。システムの観点から、カーネルゾーン自体はアプリケーションです。ARC の増加を制限して、カーネルゾーンとそれらの中で実行するアプリケーションを含むアプリケーションでより多くのメモリーを使用できるようにします。



注意 - このパラメータをホストシステムの ZFS ARC を制限するように設定できないと、メモリー不足の障害につながる可能性があります。

システムの ZFS ARC を制限するには、管理者として、大域ゾーンに `user_reserve_hint_pct` パラメータを設定します。推奨事項は、実行中のシステムで動的にパラメータを調整する `set_user_reserve.sh` と呼ばれるスクリプトを使用して、パラメータ値を 80 に設定することです。

ホストシステムで実行することが予想されるすべてのカーネルゾーンおよびその他のプロセスの最大メモリー要件に応じて、80 よりも高い値または低い値を設定できます。

`set_user_reserve.sh` スクリプトを取得し、`user_reserve_hint_pct` チューニング可能パラメータの要件の特定と構成に関する詳細を参照するには、[My Oracle Support Web](#) サイトにログインし、[Oracle Solaris 11.x の ZFS とアプリケーション間のメモリー管理 \(Doc ID 1663862.1\)](#) に関するドキュメントにアクセスします。`set_user_reserve.sh` スクリプトはそのドキュメントに添付されています。

このパラメータを設定するには、`set_user_reserve.sh` スクリプトを使用します。たとえば、`global` というシステムの大域ゾーンで:

```
global# ./set_user_reserve.sh -fp 80
Adjusting user_reserve_hint_pct from 0 to 80
Monday, March 30, 2015 04:59:47 PM PST :
waiting for current value : 60 to grow to target : 65
...
Adjustment of user_reserve_hint_pct to 80 successful.
Make the setting persistent across reboot by adding to /etc/system

#
```



```
# Tuning based on MOS note 1663861.1, script version 1.0
# added Monday, March 30, 2015 05:09:53 PM PST by system administrator : <me>
set user_reserve_hint_pct=80
```

スクリプトを実行すると、`user_reserve_hint_pct` パラメータは、実行中のシステムで調整されますが、リブート間でこれを維持させるため、`/etc/system` にパラメータを設定する必要があります。

Oracle Solaris カーネルゾーンの構成

このセクションでは、Oracle Solaris カーネルゾーンを構成する方法について説明します。

▼ カーネルゾーンを構成する方法

この手順では、4つの仮想 CPU および 4G バイトのメモリーを構成するデフォルトのカーネルゾーンテンプレート `SYSsolaris-kz` を使用して、カーネルゾーンを構成する方法について説明します。1つの仮想 CPU と 2G バイトのメモリーの最小のカーネルゾーンを構成するために、追加のテンプレート `SYSsolaris-kz-minimal` を使用できます。

ゾーンテンプレートのプロパティの概要については、『Oracle Solaris ゾーン構成リソース』の「[zonecfg template プロパティとトークン](#)」を参照してください。ゾーン構成に関する一般的な情報については、『Oracle Solaris ゾーンの作成と使用』の第1章、「[非大域ゾーンの計画および構成方法](#)」を参照してください。

始める前に ホストシステムのカーネルゾーンのハードウェアサポート、ソフトウェアサポート、およびメモリー構成を確認する必要があります。15 ページの「[カーネルゾーンホストでのハードウェアおよびソフトウェアサポートの確認](#)」および16 ページの「[カーネルゾーンのメモリーを予約するための ZFS ARC のチューニング](#)」を参照してください。

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. 新しいカーネルゾーン構成を作成します。

デフォルトの `solaris-kz` ブランドゾーンテンプレートは `SYSsolaris-kz` です。たとえば、システム `global` で、カーネルゾーン `kzone1` の新しいカーネルゾーン構成を作成するには:

```
global# zonecfg -z kzone1
```

```
Use 'create' to begin configuring a new zone.
zonecfg:kzone1> create -t SYSsolaris-kz
```

この手順の残りの構成手順では、カーネルゾーン kzone1 を使用します。

3. 追加のカーネルゾーンリソースを追加します。

カーネルゾーンリソースは、すぐに設定するかゾーンの構成後に設定できます。詳細は、[18 ページの「カーネルゾーンリソースの構成およびカスタマイズ」](#)を参照してください。

4. ゾーン構成を確定します。

```
zonecfg:kzone1> commit
```

5. zonecfg を終了します。

```
zonecfg:kzone1> exit
```

6. (オプション) ゾーン構成を確認します。

インストールの前にゾーンを確認できます。このステップを省略した場合、ゾーンのインストール時に検証が自動的に実行されます。『[Oracle Solaris ゾーンの使用](#)』の「[\(オプション\) インストール前に構成済みのゾーンを検証する方法](#)」を参照してください。

```
# zoneadm -z zonename verify
```

たとえば、システム global 上のカーネルゾーン kzone1 を検証するには:

```
global# zoneadm -z kzone1 verify
```

エラーメッセージが表示され、ゾーンの検証に失敗した場合は、メッセージに従って修正を行い、コマンドを再度実行してください。エラーメッセージが表示されない場合は、ゾーンをインストールできます。

カーネルゾーンリソースの構成およびカスタマイズ

ゾーン構成リソースにより、ゾーンのシステムリソースを管理できます。ゾーン構成の作成時にリソースを指定します。一部のリソースは、カーネルゾーンでのみ、またはネイティブゾーンでのみサポートされます。

このセクションでは、リソースを構成して、次のコンポーネントのサポートを追加する方法について説明します。

- カーネルゾーンの CPU。 [19 ページの「カーネルゾーンの CPU の管理」](#)を参照してください。
- カーネルゾーンのメモリー。 [21 ページの「カーネルゾーンのメモリーの管理」](#)を参照してください。

- カーネルゾーンのストレージデバイス。25 ページの「[カーネルゾーンのストレージデバイスとブート順序の管理](#)」を参照してください。
- カーネルゾーンのネットワークデバイスおよびネットワーク構成。26 ページの「[カーネルゾーンのネットワークデバイスと構成の管理](#)」を参照してください。
- カーネルゾーンのネットワーク仮想化。28 ページの「[カーネルゾーンでのシングルルート I/O NIC 仮想化の管理](#)」を参照してください。
- カーネルゾーンのベリファイドブート。36 ページの「[ベリファイドブートを使用した Oracle Solaris カーネルゾーンのセキュリティー保護](#)」を参照してください。
- カーネルゾーンのリソースの一時停止。34 ページの「[suspend リソースの構成](#)」を参照してください。

zonecfg コマンドを大域ゾーンで使用して、カーネルゾーンリソースを設定または変更します。

注記 - zonecfg コマンドを使用するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

ゾーンリソースの詳細については、『[Oracle Solaris ゾーン構成リソース](#)』および [solaris-kz\(5\)](#) のマニュアルページを参照してください。

カーネルゾーンの CPU の管理

デフォルトで、カーネルゾーンには作成時に 4 つの仮想 CPU が与えられます。カーネルゾーンの CPU の数を構成する次のいずれかの方法を使用して、仮想 CPU の数を変更できます。

- dedicated-cpu リソースの追加と変更
- virtual-cpu リソースの追加と変更
- anet 待機時間グループからの CPU の追加

virtual-cpu および dedicated-cpu ゾーンリソースの設定方法については、『[Oracle Solaris ゾーンの作成と使用](#)』の第 1 章、「[非大域ゾーンの計画および構成方法](#)」を参照してください。

dedicated-cpu リソースの追加

パフォーマンスを最適にするために、dedicated-cpu リソースプロパティを構成することを推奨します。この値を設定すると、カーネルゾーンは選択された CPU 上での

み実行するように指定されます。カーネルゾーンに専用の CPU で、システム上のほかのプロセスを実行することはできません。

CPU の値は、使用可能なコアまたはプロセッサの観点から割り当てることができません。システムのプロセッサ情報を取得するには、`psrinfo -vp` を使用します。たとえば、次の `psrinfo -vp` 出力は、システム `global` に 4 つの使用可能なコアがあることを示しています。

```
global# psrinfo -vp
The physical processor has 4 virtual processors (0-3)
  x86 (GenuineIntel 206D7 family 6 model 45 step 7 clock 2400 MHz)
    Intel(r) Xeon(r) CPU E5-2609 0 @ 2.40GHz
```

注記 - デフォルトで、`dedicated-cpu:ncpus` を設定しても、システムのどの CPU が割り当てられるかを制御しません。このため、システムがリブートした場合に、不整合な結果となる可能性があります。`dedicated-cpu:cpus` を使用して、使用する正確な CPU を指定してください。詳細については、『[Oracle Solaris ゾーン構成リソース](#)』の「[dedicated-cpu リソース](#)」を参照してください。

`dedicated-cpu` ゾーンリソースに関する全般的な情報については、『[Oracle Solaris ゾーンの作成と使用](#)』の第 1 章、「[非大域ゾーンの計画および構成方法](#)」を参照してください。

例 1 カーネルゾーンへの専用 CPU の追加

この例は、専用 CPU をカーネルゾーン `kzone1` に追加する方法を示しています。

```
global# zonecfg -z kzone1
zonecfg:kzone1> info dedicated-cpu
zonecfg:kzone1> add dedicated-cpu
zonecfg:kzone1:dedicated-cpu> set ncpus=8
zonecfg:kzone1:dedicated-cpu> end
zonecfg:kzone1> info dedicated-cpu
      ncpus: 8
zonecfg:kzone1> exit
```

virtual-cpu リソースの追加

`virtual-cpu` リソースは、カーネルゾーンで可視になる仮想 CPU の数を指定します。ホスト上では、仮想 CPU は CPU 時間をほかのゾーンと共有します。`virtual-cpu` リソースを設定することは、統合に役立ちますが、システムのパフォーマンスに影響する可能性があります。

`dedicated-cpu` リソースをすでに定義している場合、構成されている仮想 CPU のデフォルトの数は、`dedicated-cpu` リソース内の `ncpus` 範囲の下限値と一致します。両方のリソースが存在する場合は、整合性が相互チェックされます。詳細は、[zonecfg\(1M\)](#) のマニュアルページを参照してください。

例 2 カーネルゾーンへの仮想 CPU の追加

この例は、`virtual-cpu` リソースを使用してカーネルゾーン `kzone1` に追加の仮想 CPU を追加する方法を示しています。

```
global# zonecfg -z kzone1
zonecfg:kzone1> info virtual-cpu
zonecfg:kzone1> add virtual-cpu
zonecfg:kzone1:virtual-cpu> set ncpus=8
zonecfg:kzone1:virtual-cpu> end
zonecfg:kzone1> info virtual-cpu
virtual-cpu:
    ncpus: 8
zonecfg:kzone1> exit
```

待機時間グループからの CPU の追加

待機時間グループから CPU を指定できます。待機時間グループから CPU を指定すると、待機時間のグループがベースとなるネットワークデバイスと同じ場合に、ネットワークパフォーマンスを向上させることができます。

待機時間グループの操作の詳細については、『[Oracle Solaris 11.3 での仮想ネットワークとネットワークリソースの管理](#)』の第 2 章、「[仮想ネットワークの作成および管理](#)」を参照してください。

カーネルゾーンのメモリーの管理

一定量の物理 RAM をカーネルゾーンの仮想プラットフォームに割り当てる必要があります。この量を定義するには、カーネルゾーンの `capped-memory` リソースの `physical` プロパティを設定します。

カーネルゾーンに割り当てられる物理メモリーは、ゾーンのブート時にその全体に割り当てられます。割り当てられるメモリーは、カーネルゾーンの排他的使用のためです。カーネルゾーンがブートしたら、`capped-memory` リソースに指定されたすべてのメモリーがホストオペレーティングシステムに使用されているように見えます。

デフォルトのカーネルゾーンのメモリーサイズ (`capped-memory:physical`) は 4G バイト (GB) です。より大きいワークロードを管理するには、メモリーサイズを増やすことをお勧めします。

カーネルゾーンのデフォルトの CPU およびメモリー構成は、アプリケーションの実行を容易にするために、4 つの VCPU と 4G バイトのメモリーです。追加のカーネルゾーンテンプレート `SYSsolaris-kz-minimal` は 1 VCPU と 2G バイトのメモリーのサポートされている最小のカーネルゾーンの構成を提供します。

x86 ベースシステムでは、2M バイト (MB) の増分で `capped-memory` リソースを設定する必要があります。

SPARC ベースシステムでは、256M バイト (MB) の増分で `capped-memory` リソースを設定する必要があります。

ゾーンは、ゾーンのブート時に `capped-memory` リソースを割り当てます。この量は、ゾーンの実行中は固定されたままになります。

`capped-memory:pagesize-policy` プロパティは、カーネルゾーンの物理メモリのページサイズを割り当てるためのポリシーを指定します。デフォルトでは、カーネルゾーンは最適なパフォーマンスを実現するために使用可能な最大のページサイズを使用します。詳細は、[23 ページの「メモリのページサイズポリシーと物理メモリについて」](#)を参照してください。

注記 - ゾーンテンプレート `SYSsolaris-kz-minimal` は、サポートされる最小カーネルゾーン構成である 1 個の VCPU および 2G バイトのメモリを提供します。Fujitsu M10 または SPARC M10 サーバーでは、このテンプレートで作成されたカーネルゾーンが、メモリ不足のためにブートできない可能性があります。カーネルゾーンをブートできない場合は、`capped-memory` リソースの `physical` プロパティを使用して、カーネルゾーンに割り当てられるメモリを増やしてください。

`capped-memory` ゾーンリソースの設定方法については、『[Oracle Solaris ゾーン作成と使用](#)』の第 1 章、「[非大域ゾーンの計画および構成方法](#)」を参照してください。

`capped-memory` ゾーンリソースの設定の詳細は、『[Oracle Solaris ゾーン構成リソース](#)』の「[solaris-kz ゾーンと capped-memory リソース](#)」を参照してください。

インストール前にカーネルゾーンのメモリサイズを増やした場合、スワップデバイスとダンプデバイスも大きくなるため、カーネルゾーンのルートディスクサイズも増やす必要があります。カーネルゾーンにディスクを明示的に追加しない場合、`zvol` が作成され、ルートディスクとして使用されます。デフォルトで、`zvol` のサイズは 16G バイトです。異なるルートディスクサイズが必要な場合は、`zoneadm install -x install-size` コマンドを使用して、ディスクサイズを変更します。たとえば、カーネルゾーン `kzone1` に 32G バイトのルートディスクサイズを指定するには、次のインストール時に、次のコマンドを使用します。

```
global# zoneadm -z kzone1 install -x install-size=32G
```

`zoneadm` コマンドを使用したディスクサイズの変更については、[zoneadm\(1M\)](#) のマニュアルページを参照してください。

例 3 SPARC ベースシステムでの `capped-memory` リソースの設定

この例では、SPARC ベースシステムに、`capped-memory` リソースの `physical` プロパティを設定して、2048M バイトのメモリを指定する方法を示します。

```
global# zonecfg -z kzone1
zonecfg:kzone1> select capped-memory
zonecfg:kzone1:capped-memory> set physical=2048m
zonecfg:kzone1:capped-memory> end
zonecfg:kzone1> exit
```

例 4 x86 システムでの capped-memory リソースの設定

この例では、x86 システムに、capped-memory リソースの physical プロパティを設定して、16G バイトのメモリーを指定する方法を示します。

```
global# zonecfg -z kzone1
zonecfg:kzone1> select capped-memory
zonecfg:kzone1:capped-memory> set physical=16g
zonecfg:kzone1:capped-memory> end
zonecfg:kzone1> exit
```

メモリーのページサイズポリシーと物理メモリーについて

capped-memory リソースの pagesize-policy プロパティは、システムでカーネルゾーンのページサイズを選択する方法を制御します。

デフォルトのカーネルゾーンテンプレート SYSsolaris-kz では、pagesize-policy プロパティを、最適なパフォーマンスを得るための推奨値である largest-available に設定します。この設定により、システムはカーネルゾーンの物理メモリー量に合わせて使用する適切なページサイズを選択できます。物理メモリーサイズはページサイズの倍数である必要があるため、システムではカーネルゾーン用に指定された物理メモリー量に合致する最大ページサイズを選択します。pagesize-policy=largest-available を使用したブートは常に成功します。

最適なパフォーマンスを得るには、pagesize-policy=largest-available の設定時に最大ページサイズの選択が可能になるように、適切な量の物理メモリーを設定します。

physical プロパティを、サポートされている最大ページサイズの倍数になる量に設定する必要があります。割り当てられているメモリーの量が要求されているページサイズと完全に一致する必要があります。次の例を参照してください。

例 5 最大ページサイズを使用するように物理メモリーを設定する

SPARC T5 システムでは、さまざまなページサイズがサポートされていることが次の出力でわかります。最大ページサイズは 2147483648 バイト (2G バイト) です。

2147483648 ページサイズを使用するには、pagesize-policy=largest-available の設定時に最大ページサイズを使用できるように、capped-memory:physical プロパティを 2G バイトの倍数値である 8G バイトに設定します。

```
global# pagesize -a
```

```

8192
65536
4194304
268435456
2147483648
global# zonecfg -z kzone1
zonecfg:kzone1> select capped-memory
zonecfg:kzone1:capped-memory> set physical=8G
zonecfg:kzone1:capped-memory> info
capped-memory:
  physical: 8G
  pagesize-policy: largest-available
zonecfg:kzone1:capped-memory> end
zonecfg:kzone1> exit

```

カーネルゾーンの `pagesize-policy` プロパティがクリアされているか、設定されていない場合、カーネルゾーンではそれが実行されている特定のハードウェアプラットフォームでブートするために必要な許容最小ページサイズを使用します。

Oracle Solaris 11.3 のアップデートで作成されたカーネルゾーンを、`pagesize-policy` プロパティをサポートしていない古い Oracle Solaris リリース (Oracle Solaris 11.3 の初期リリースなど) に移行する場合は、`pagesize-policy` をクリアする必要があります。次の例を参照してください。

例 6 以前のバージョンの Oracle Solaris への移行前に `pagesize-policy` プロパティをクリアする

この例では、`pagesize-policy` プロパティをクリアして、そのプロパティをサポートしていない古いリリースの Oracle Solaris にカーネルゾーンを移行する準備を整える方法を示します。

```

global# zonecfg -z kzone1
zonecfg:kzone1> select capped-memory
zonecfg:kzone1:capped-memory> clear pagesize-policy
zonecfg:kzone1:capped-memory> end
zonecfg:kzone1> commit

```

最初に、移行の試験実行を行い、`zoneadm migrate -n` オプションを使用して `pagesize-policy` をクリアする必要があるかどうかを確認できます。詳細は、`zoneadm(8)` のマニュアルページまたは [第3章「Oracle Solaris カーネルゾーンの移行」](#) を参照してください。

まれに、最大ページサイズを使用していなければ、ブートを回避するために `pagesize-policy=largest-only` を設定することが必要な場合もあります。これは、パフォーマンスが最大ページサイズの使用に左右されるデータベースまたはほかのアプリケーションをホストするカーネルゾーンに役立つことがあります。

`physical` および `pagesize-policy` プロパティの設定方法の詳細は、『[Oracle Solaris ゾーン構成リソース](#)』の「[solaris-kz ゾーンと capped-memory リソース](#)」を参照してください。

カーネルゾーンのストレージデバイスとブート順序の管理

カーネルゾーンのルートは常にアクセス可能です。デフォルトでは、カーネルゾーンのインストールは、ルートディスクに 16G バイトの ZFS ボリュームを使用します。-x `install-size` オプションを付けて `zoneadm -z install` コマンドを使用して、ゾーンのインストール時に異なるサイズを指定できます。たとえば、カーネルゾーン `kzone1` の ZFS のサイズを 32G バイトに増やすには、次のようにします。

```
global# zoneadm -z kzone1 install -x install-size=32g
```

`add device` リソースを使用すると、追加のストレージデバイスをカーネルゾーンに追加できます。デバイスは、システム間で移植可能で、ZFS ボリュームのパフォーマンスを向上させます。追加のカーネルゾーンのストレージデバイスには、次の要件があります。

- ストレージデバイスのフルパス (`/dev/rdisk/c9t0d0` など) を指定する必要があります。
- ストレージデバイスは次のいずれか 1 つで定義する必要があります。
 - `add device match` リソースプロパティ。 `add device match` リソースプロパティにストレージデバイスを指定する場合は、`/dev/rdisk`、`/dev/zvol/rdisk`、または `/dev/did/rdisk` に存在するデバイスを指定する必要があります。
 - 有効なストレージ URI。
- ストレージデバイスはディスク全体か LUN である必要があります。

ストレージデバイスごとのブート順序を指定するには、`bootpri` リソースプロパティを使用します。`bootpri` リソースプロパティは正の整数値に設定する必要があります。



注意 - デバイスをブートデバイスとして使用する場合のみ、`bootpri` リソースプロパティを設定する必要があります。`bootpri` リソースプロパティをブートデバイス以外のデバイスに設定すると、データが破損する可能性があります。

`bootpri` リソースプロパティの設定を解除するには、`clear bootpri` リソースプロパティを使用します。

ブート可能な複数のデバイスがインストール中に存在する場合、これらのデバイスはゾーン内のミラー化 ZFS プールで使用されます。

デバイスごとのデフォルトのブート順は、最初に `bootpri` でデバイスをソートし、複数のデバイスが同じ `bootpri` を持つ場合は、次に `id` でソートして決定されます。

例 7 カーネルゾーンへのストレージデバイスの追加

この例は、ストレージデバイス /dev/rdisk/c9t0d0 をカーネルゾーン kzone1 に追加する方法を示しています。

```
global# zonecfg -z kzone1
zonecfg:kzone1> add device
zonecfg:kzone1:device> set match=/dev/rdisk/c9t0d0
zonecfg:kzone1:device> set bootpri=4
zonecfg:kzone1:device> end
```

例 8 ストレージ URI を使用するためのカーネルゾーンのデフォルトのブートデバイスの変更:

この例は、iscsi://zfssa/lunname.naa.600144F0DBF8AF19000052E820D60003 にあるストレージ URI を使用するように、カーネルゾーン kzone1 でデフォルトのブートデバイスを変更する方法を示しています。

```
global# zonecfg -z kzone1
zonecfg:kzone1> select device id=0
zonecfg:kzone1:device> set storage=iscsi://zfssa/lunname.naa.600144F0DBF8AF19000052E820D60003
zonecfg:kzone1:device> end
zonecfg:kzone1> info device
device:
  match not specified
  storage: iscsi://zfssa/lunname.naa.600144F0DBF8AF19000052E820D60003
  id: 0
  bootpri: 0
```

カーネルゾーンのネットワークデバイスと構成の管理

カーネルゾーンは、net または anet リソースの追加によってカーネルゾーンでのネットワークアクセスを提供します。これらの2つのリソースタイプの詳細は、『Oracle Solaris ゾーン構成リソース』の「構成可能なリソースとプロパティ」および『Oracle Solaris ゾーン構成リソース』を参照してください。

注記 - カーネルゾーンでは anet リソースを使用することをお勧めします。

排他的 IP ゾーンをカーネルゾーンに使用する必要があります。排他的 IP ゾーンの詳細は、『Oracle Solaris ゾーンの作成と使用』の「排他的 IP ゾーンのネットワークアドレス」を参照してください。

追加の MAC アドレスを指定すると、入れ子にされたゾーン、またはカーネルゾーンが非大域 solaris および solaris10 ブランドゾーンをホストするゾーンをサポートできます。入れ子にされたゾーンの詳細は、70 ページの「入れ子にされたゾーンの管理」を参照してください。

オプションでネットワークデバイス ID を指定して、ゾーン内から VNIC アドレスを識別して、ネットワークインタフェースがカーネルゾーンに表示される順序を決定でき

ます。このプロセスは、ある物理スロットから別の物理スロットへの NIC の移動に似ています。

ネットワークゾーンリソースの設定方法については、『[Oracle Solaris ゾーンの作成と使用](#)』の第 1 章、「[非大域ゾーンの計画および構成方法](#)」を参照してください。

例 9 カーネルゾーンへのネットワークリソースの追加

この例は、ネットワークリソースをカーネルゾーン `kzone1` に追加する方法を示しています。ID が 3 に設定され、新しい `anet` インタフェースがカーネルゾーンに表示される順序が決定されます。

```
global# zonecfg -z kzone1
zonecfg:kzone1> add anet
zonecfg:kzone1:anet> set id=3
zonecfg:kzone1:anet> end
zonecfg:kzone1> exit
```

例 10 カーネルゾーンからのネットワークデバイスの削除

この例は、カーネルゾーン `kzone1` からネットワークデバイスを削除する方法を示しています。既存の `anet` リソースに関する情報が一覧表示され、値が 1 の `anet` デバイスが削除されます。

```
global# zonecfg -z kzone1 info anet
anet:
  lower-link: auto
  allowed-address not specified
  allowed-dhcp-cids not specified
  link-protection: mac-nospoof
  mac-address: random
  mac-prefix not specified
  mac-slot not specified
  vlan-id not specified
  priority not specified
  rxrings not specified
  txrings not specified
  mtu not specified
  maxbw not specified
  rxfanout not specified
  vsi-typeid not specified
  vsi-vers not specified
  vsi-mgrid not specified
  etsbw-lcl not specified
  cos not specified
  id: 0
anet:
  lower-link: auto
  allowed-address not specified
  allowed-dhcp-cids not specified
  link-protection: mac-nospoof
  mac-address: default
  mac-prefix not specified
  mac-slot not specified
  vlan-id not specified
  priority not specified
  rxrings not specified
```

```
txrings not specified
mtu not specified
maxbw not specified
rxfanout not specified
vsi-typeid not specified
vsi-vers not specified
vsi-mgrid not specified
etsbw-lcl not specified
cos not specified
id: 1
global# zonecfg -z kzone1 remove anet id=1
```

カーネルゾーンでのシングルルート I/O NIC 仮想化の管理

zonecfg iov anet プロパティを使用して、カーネルゾーンでシングルルート I/O (SR-IOV) NIC 仮想機能 (VF) を作成および管理できます。SR-IOV は仮想マシン間での PCIe (Peripheral Component Interconnect Express) デバイスの効率的な共有を可能にし、ネイティブのパフォーマンスに匹敵する I/O パフォーマンスを実現できるようなシステムハードウェアに実装されています。Oracle Solaris での SR-IOV の使用については、『[Oracle Solaris 11.3 での仮想ネットワークとネットワークリソースの管理](#)』の「[VNIC でのシングルルート I/O 仮想化の使用](#)」を参照してください。

zonecfg iov プロパティはカーネルゾーンでのみサポートされています。ネイティブな solaris ゾーンでのサポートは提供されません。

zonecfg iov anet プロパティの有効化および構成の方法については、『[Oracle Solaris ゾーン構成リソース](#)』の「[リソースタイプとプロパティ](#)」を参照してください。

▼ 単一の anet を使用してカーネルゾーンで SR-IOV NIC 仮想機能を有効にする方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. anet の iov を有効にします。

zonecfg を使用して、選択した anet の iov を有効にします。

```
# zonecfg -z kernel-zone
zonecfg:kernel-zone> set lower-link=network-interface
zonecfg:kernel-zone> select anet id=id-number
zonecfg:kernel-zone:anet> set lower-link=network-interface
zonecfg:kernel-zone:anet> set iov=iov-value
```

```
zonecfg:kernel-zone:anet> end
zonecfg:kernel-zone exit
```

次の例は、カーネルゾーン kzone1 に属する anet の iov を有効にする方法を示しています。

```
global# zonecfg -z kzone1
zonecfg:kzone1> select anet id=0
zonecfg:kzone1:anet> set lower-link=net1
zonecfg:kzone1:anet> set iov=auto
zonecfg:kzone1:anet> end
zonecfg:kzone1> exit
```

3. (オプション) カーネルゾーン構成で、anet に iov プロパティが設定されていることを確認します。

```
# zonecfg -z kernel-zone info anet id=id-number
```

たとえば、システム global およびカーネルゾーン kzone1 の anet 0 の場合:

```
# zonecfg -z kzone1 info anet id=0
anet:
  lower-link: net1
  allowed-address not specified
  configure-allowed-address: true
  ...
  iov: auto
  lro: auto
  id: 0
```

4. dladm(1M) を使用して、選択したネットワークインタフェース上で SR-IOV が有効になっていることを確認します。

```
# dladm show-linkprop -p iov network-interface
```

たとえば、システム global およびネットワークインタフェース net1 の場合:

```
global# dladm show-linkprop -p iov net1
LINK  PROPERTY  PERM VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net1  iov          rw on      on         auto    auto,on,off
```

5. カーネルゾーンをブートします。

```
# zoneadm -z kernel-zone boot
```

たとえば、システム global でカーネルゾーン kzone1 をブートするには:

```
global# zoneadm -z kzone1 boot
```

6. VF が正常に追加されたことを確認します。

```
# zlogin kernel-zone
kernel-zone# dladm show-phys
```

例:

```
global# zlogin kzone1
kzone1# dladm show-phys
LINK  MEDIA  STATE  SPEED  DUPLEX  DEVICE
net0  Ethernet  down  0      unknown ixgbev0
```

例 11 anet の zonecfg iov 値の確認

次の例は、anet 0 の iov 値を示しています。値は、デフォルト値である off に設定されています。

```
global# zonecfg -z kzone1
zonecfg:kzone1> select anet id=0
zonecfg:kzone1:anet> info
anet:
    lower-link: net1
    allowed-address not specified
    configure-allowed-address: true
    ...
    iov: off
    lro: auto
    id: 0
zonecfg:kzone1:anet> end
zonecfg:kzone1> exit
```

例 12 anet での iov および VLAN タグ付けの構成

次の例は、anet 上で VLAN タグ付けを有効にするよう VLAN ID を明示的に設定する方法を示しています。これにより、タグ付けされていない、悪質な可能性のあるフレームを破棄できます。

```
global# zonecfg -z kzone1
zonecfg:kzone1> select anet id=0
zonecfg:kzone1:anet> set iov=auto
zonecfg:kzone1:anet> set vlan-id=11
zonecfg:kzone1:anet> end
zonecfg:kzone1> exit
```

Oracle Solaris カーネルゾーンでの仮想機能とシャドウ VNIC の使用

カーネルゾーンの仮想機能は、カーネルゾーンに属する anet の zonecfg iov プロパティが on または auto に設定されているときに作成されます。VF はホストシステムによってカーネルゾーンに割り当てられます。

カーネルゾーンに割り当てられた各 VF には、ホスト内のシャドウ VNIC が関連付けられます。シャドウ VNIC を使用してネットワーク統計情報を表示できます。

次にシステム global のシャドウ VNIC kzone1/net0 の出力例を示しています。

```
global# dladm show-link
LINK          CLASS      MTU      STATE    OVER
net1          phys       1500     unknown --
net0          phys       1500     up       --
net2          phys       1500     up       --
kzone1/net0   vnic       1500     unknown net1

global# dlstat show-link kzone1/net0
LINK          IPKTS     RBYTES   OPKTS    OBYTES
```

```
kzone1/net0      0      0      3      126
```

シャドウ VNIC はデータを転送できないため、シャドウ VNIC を DLMP またはトランクアグリゲーションに使用することはできません。さらに、シャドウ VNIC のリンクプロパティを構成することはできません。

zonecfg anet プロパティ bwshare を使用すると、ベースとなる物理リンクがサポートされている場合にのみ、リンク上にシャドウ VNIC を設定できます。詳細については、[dladm\(1M\)](#) および [zonecfg\(1M\)](#) のマニュアルページを参照してください。

VNIC とネットワーク構成の詳細については、『[Oracle Solaris 11.3 での仮想ネットワークとネットワークリソースの管理](#)』を参照してください。

カーネルゾーンでの仮想 LAN の構成

Ethernet ベースの anet を使用して、カーネルゾーンの内地に VNIC を作成し、それらが独自の仮想 LAN (VLAN) に含まれるように構成できます。

vlan リソースを使用すると、既存の anet リソースに VLAN ID (VID) を追加して、新しい VLAN を作成できます。anet および vlan リソースの詳細は、『[Oracle Solaris ゾーン構成リソース](#)』の「[構成可能なリソースとプロパティ](#)」を参照してください。

vlan リソースは、カーネルゾーンを VLAN 対応のものにします。ホストシステムでは、これらの VLAN 用のパケットを、VLAN タグを削除せずにカーネルゾーンに転送します。その後、カーネルゾーンではそのパケットを正しいネットワーククライアントに転送します。

データの送信時に、これらの VLAN からのパケットはカーネルゾーンによってタグが付けられて、ホストに渡されます。ホストでは、宛先 MAC に基づいて、タグを削除せずにパケットを転送します。

注記 - anet 用の VLAN を追加する前に、anet 用の vlan-id (ポート VID または PVID と呼ばれる) を指定する必要はありません。PVID が設定されていない場合は、ゾーンの MAC アドレスと一致する、タグの付いていないすべてのパケットがホストのゾーンに渡されます。

例 13 追加の VLAN を使用したカーネルゾーンの構成

0:1:2:3:4:5 という MAC アドレス、11 という PVID、および 45 と 46 という 2 つの追加の VID を使用して、kz0 を構成します。

```
global# zonecfg -z kz0
zonecfg:kz0> create -t SYSsolaris-kz
zonecfg:kz0> select anet id=0
```

```
zonecfg:kz0:anet> set mac-address=0:1:2:3:4:5
zonecfg:kz0:anet> set vlan-id=11
zonecfg:kz0:anet> add vlan
zonecfg:kz0:anet:vlan> set vlan-id=45
zonecfg:kz0:anet:vlan> end
zonecfg:kz0:anet> add vlan
zonecfg:kz0:anet:vlan> set vlan-id=46
zonecfg:kz0:anet:vlan> end
zonecfg:kz0:anet> info vlan
vlan 0:
    vlan-id: 45
vlan 1:
    vlan-id: 46
zonecfg:kz0:anet> end
zonecfg:kz0> commit
zonecfg:kz0> exit
```

ゾーンがインストールされてブートされたあと、`dladm show-vnic` コマンドによって次が表示されます。

```
global# dladm show-vnic
LINK          OVER          SPEED  MACADDRESS  MACADDRTYPE  IDS
kz0/net0      net4           10000  0:1:2:3:4:5  fixed        VID:11,45,46
```

これで、ホストシステム `global` の仮想スイッチが、次の `<mac-address, vlan-id>` タブルを使用してフレームを処理するように構成されました。

- `<0:1:2:3:4:5, 11>`
- `<0:1:2:3:4:5, 45>`
- `<0:1:2:3:4:5, 46>`

`<0:1:2:3:4:5, 11>` タブルで到着したフレームはシステム `global` によって VID が削除されて、カーネルゾーン `kz0` に渡されるため、`kz0` では VID 11 でタグ付けされたパケットを認識することはありません。`<0:1:2:3:4:5, 45>` と `<0:1:2:3:4:5, 46>` が指定されたフレームは、それぞれ VID 45 と VID 46 のタグが付いたまま `kz0` に渡されます。

VID が 45 の VLAN データリンク `vlan45` が `kz0` 内にある場合、`kz0` の仮想スイッチはフレームから VID 45 を削除して、フレームを `vlan45` に渡します。`kz0` 内の `vlan45` データリンクから発信されるすべてのフレームは、`kz0` の仮想スイッチによってタグ付けされ、ホストの `anet` に渡されます。ホスト `anet` は、送信する NIC にフレームを直接渡します。

例 14 カーネルゾーンでサポートされている VLAN ID の一覧の表示

カーネルゾーンの内側で、`dladm show-phys -v` を使用して、物理データリンクでサポートされている VLAN ID を特定します。

```
global# zlogin kz0
kz0# dladm show-phys -v
LINK  VID  INUSE  CLIENT
net0   40   yes    vnic0,vnic1
       20   no     --
       15   yes    vnic2
net1   32   no     --
       11   no     --
```


10 no --

カーネルゾーンでの動的な MAC アドレスと VLAN ID の使用

ほとんどの配備の場合、カーネルゾーンで使用される MAC アドレスと VLAN ID は、ゾーンをブートする前に静的に構成できます。ただし、場合によっては、カーネルゾーンでその VNIC の MAC アドレスと VLAN ID に使用する必要のある値が前もってわからないことがあります。その場合、カーネルゾーンでそのブート時に使用する必要のある MAC アドレスと VLAN ID をホストに通知できるようにするために、許可される MAC アドレスの接頭辞と許可される VLAN ID の範囲を指定できます。カーネルゾーンで有効な任意の MAC アドレスまたは VLAN ID を使って VNIC を作成できるようにすることも可能です。

注記 - MAC アドレスと VLAN ID の数、およびそれらの値が前もってわかる場合は、デフォルトの静的構成を使用するようにしてください。静的構成は、SR-IOV VF ベースの anet にも必要です。

動的構成を有効にするには、次の手順に示すように、anet プロパティー allowed-mac-address および allowed-vlan-ids を設定します。

これらのプロパティーの詳細は、『[Oracle Solaris ゾーン構成リソース](#)』の「[リソースタイプのプロパティー](#)」を参照してください。

▼ 動的な MAC アドレスおよび VLAN ID をカーネルゾーンの anet 構成に使用する方法

1. 管理者になります。
詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。
2. anet の allowed-mac-address を有効にします。
zonecfg を使用して、anet デバイスおよび mac リソースを追加し、それに対して allowed-mac-address を有効にします。

```
# zonecfg -z kernel-zone
zonecfg:kernel-zone> add anet
zonecfg:kernel-zone:anet> add mac
zonecfg:kernel-zone:anet:mac> add allowed-mac-address octet-prefix

zonecfg:kernel-zone:anet:mac> end
zonecfg:kernel-zone:anet>
```

3. anet の dynamic-vlan-id を有効にします。

zonecfg を使用して、vlan リソースを追加し、それに対して allowed-vlan-ids を有効にします。

```
zonecfg:kernel-zone:anet> add vlan
zonecfg:kernel-zone:anet:vlan> add allowed-vlan-ids id-range
zonecfg:kernel-zone:anet:vlan> end
zonecfg:kernel-zone:anet> end
zonecfg:kernel-zone> exit
```

4. カーネルゾーンをブートします。

```
# zoneadm -z kernel-zone boot
```

5. カーネルゾーンにログインします。

```
# zlogin kernel-zone
```

6. カーネルゾーンで動的なアドレスと ID を確認します。

許可されている MAC 接頭辞と VLAN ID を特定するには、dladm show-phys コマンドで -o オプションを使用します。

```
# dladm show-phys -o link,media,device,allowed-addresses,allowed-vids
```

たとえば、kzone1 と呼ばれるゾーンで確認するには:

```
global# zlogin kzone1
kzone1# dladm show-phys -o link,media,device,allowed-addresses,allowed-vids
LINK  MEDIA      DEVICE  ALLOWED-ADDRESSES  ALLOWED-VIDS
net0  Ethernet    zvnet0  fa:16:3f,          100-199,
                                fa:80:20:21:22    400-498,500
```

IPoIB とカーネルゾーンの操作

InfiniBand (IPoIB) デバイスをサポートするようにカーネルゾーンを構成するには、anet リソースのプロパティを設定します。『Oracle Solaris ゾーン構成リソース』の「リソースタイプとプロパティ」および『Oracle Solaris 11.3 での仮想ネットワークとネットワークリソースの管理』の「カーネルゾーンでの準仮想化 IPoIB データリンクの作成および表示」を参照してください。

suspend リソースの構成

カーネルゾーンの構成に suspend リソースプロパティがある場合のみ、カーネルゾーンの一時停止および再開がサポートされます。カーネルゾーンを一時停止する前に、suspend リソースを追加し、その path または storage プロパティを設定する必要があります。

一時停止および再開は、ウォーム移行に必要です。ウォーム移行を実行する場合は、ソースホストおよびターゲットホストからアクセスできる共有ストレージロケーションを suspend リソースで使用する必要があります。

そのほかに、システム保守が必要なときにゾーンを停止処理するのではなくゾーンを一時停止する機能を有効にする場合にも一時停止および再開が使用されます。一時停止および再開によって、カーネルゾーンとその実行中のアプリケーションがより迅速に使用できる状態になります。

また、`autoshtutdown=suspend` プロパティを設定することで、大域ゾーンが停止処理されるときにカーネルゾーンが停止処理されるのではなく自動的に一時停止されるようにすることもできます。`autoshtutdown` の詳細は、『Oracle Solaris ゾーン構成リソース』の「リソースタイプとプロパティ」または [zonecfg\(1M\)](#) を参照してください。

例 15 カーネルゾーンの一時停止を有効にするように suspend リソースを構成する

この例では、ホスト上でカーネルゾーンを一時停止できるようにするために、ローカルパスを使用して一時停止および再開を有効にするように suspend リソースを設定する方法を示します。

```
global# zonecfg -z kz1
zonecfg:kz1> add suspend
zonecfg:kz1:suspend> set path=/system/zones/kz1/suspend
zonecfg:kz1:suspend> end
zonecfg:kz1> info suspend
suspend:
    path: /system/zones/kz1/suspend
zonecfg:kz1> exit
```

次のコマンドを使用してゾーンを一時停止し、あとで `zoneadm boot` コマンドを使用して再開できます。

```
global# zoneadm -z kz1 suspend
```

例 16 ウォーム移行を有効にするように suspend リソースを構成する

この例では、suspend リソースをリセットして、ストレージ URI を iSCSI デバイスに使用する方法を示します。

```
global# zonecfg -z kz1
zonecfg:kz1> select suspend
zonecfg:kz1:suspend> clear path
zonecfg:kz1:suspend> set storage=iscsi://system/luname.naa.501337600144f0dbf8af1900
zonecfg:kz1:suspend> end
zonecfg:kz1> exit
```

詳細は、54 ページの「ウォーム移行を使用したカーネルゾーンの移行」を参照してください。

suspend リソースプロパティの要件の詳細は、[solaris-kz\(5\)](#) のマニュアルページを参照してください。

ベリファイドブートを使用した Oracle Solaris カーネルゾーンのセキュリティー保護

ベリファイドブートを使用すると、カーネルゾーンのブートプロセスをセキュリティー保護できます。ベリファイドブートは、Oracle Solaris カーネルモジュールを実行前にセキュアにロードすることで、破損したカーネルゾーンモジュール、悪意のあるプログラム、および未承認のサードパーティーカーネルモジュールのインストールからカーネルゾーンを保護します。

ベリファイドブートでは、次のアクションを実行できます。

- Oracle Solaris カーネルモジュールの `elfsign(1)` 検証を自動化します。デフォルトでは、Oracle Solaris システム証明書のみを検証に使用します。ベリファイドブートでは、追加の証明書を指定できるため、サードパーティーのカーネルモジュールや、別のバージョンの Oracle Solaris 向けに署名されたモジュールをロードできます。
- カーネルゾーンのリブートからブートプロセスの完了までのブートプロセスに、検証可能な信頼チェーンを作成します。

カーネルゾーンのベリファイドブートを有効化および構成するには、`verified-boot zonecfg` リソースプロパティを使用します。

ベリファイドブートと `verified-boot` リソースプロパティは、`solaris-kz` ブランドゾーンでのみサポートされます。

証明書検証および Oracle Solaris 11.3 でのベリファイドブートの詳細については、`elfsign(1)` のマニュアルページおよび『[Oracle Solaris 11.3 でのシステムおよび接続されたデバイスのセキュリティー保護](#)』の「[ベリファイドブートの使用](#)」を参照してください。

verified-boot リソースプロパティおよび elfsign 検証について

`verified-boot` リソースプロパティは、カーネルゾーンのブートポリシーおよび証明書設定を制御します。このリソースのプロパティは次のとおりです。

- `policy`
`policy` プロパティは `unix`、`genunix`、およびその他のカーネルモジュールの検証を制限します。このプロパティの指定可能な値は、次のとおりです。

<code>warning</code>	<code>elfsign</code> 検証が失敗した場合は、警告メッセージを出力します。これがデフォルト値です。
----------------------	--

none elfsign 署名検証が失敗した場合、アクションは発生しません。

注記 - policy の値が none に設定されている場合、verified-boot リソースプロパティは有効になっていません。

enforce elfsign 署名検証が失敗した場合は、警告メッセージを出力します。カーネルモジュールはロードされません。

■ cert

cert プロパティはシステム上の **elfsign(1)**X.509 公開鍵証明書の場所を指定します。証明書の場所は、X.509 証明書ファイルの URI で指定します。ローカルファイルの場合、証明書は大域ゾーンのファイルシステム内にある必要があります。リモート URI の場合、URI は大域ゾーンからアクセスできる必要があります。

新しい証明書を追加するには、add サブコマンドを使用します。最大 7 個の証明書を各カーネルゾーンに追加できます。

例 17 カーネルゾーンのベリファイドブートの有効化

この例では、システム global にカーネルゾーン kz1 を作成しています。verified-boot の policy の値は enforce に設定されます。これはカーネルに対し、ブートファイルの署名検証に失敗した場合はブートせず、失敗に関するエラーメッセージを出力するよう指示します。

```
global# zonecfg -z kz1
kz1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:kz1> create -t SYSsolaris-kz
zonecfg:kz1> set zonpath=/rpool/zones/kz1
zonecfg:kz1> set autoboot=true
zonecfg:kz1> add verified-boot
zonecfg:kz1:verified-boot> set policy=enforce
zonecfg:kz1:verified-boot> end
zonecfg:kz1> verify
zonecfg:kz1> commit
zonecfg:kz1> exit
```

例 18 複数の証明書によるカーネルゾーンのベリファイドブートの構成

この例は、システム global 上にすでに構成されているカーネルゾーン kz2 に verified-boot zonecfg リソースを追加する方法を示しています。2 つの証明書が構成に追加されます。

```
global# zonecfg -z kz2
zonecfg:kz2> add verified-boot
zonecfg:kz2:verified-boot> set policy=warning
zonecfg:kz2:verified-boot> add cert file:///etc/certs/SOLARIS-KZ
```

```
zonecfg:kz2:verified-boot> add cert http://example/keydist/cert.pem
zonecfg:kz2:verified-boot> info
verified-boot:
  policy: warning
  cert: file:///etc/certs/SOLARIS-KZ
  cert: http://example/keydist/cert.pem
zonecfg:kz2:verified-boot> end
zonecfg:kz2> verify
zonecfg:kz2> commit
```

カーネルゾーンでの Silicon Secured Memory の有効化

Silicon Secured Memory (SSM) 機能は、SPARC T7 および SPARC M7 サーバーで始まる SPARC ベースシステムで使用できます。SSM は、Application Data Integrity (ADI) と呼ばれることもあります。カーネルゾーンの以前のシステムからの移行や以前のシステムへの移行を支援するために、SSM はホストシステムで利用可能であっても、カーネルゾーン内ではデフォルトで無効になっています。

カーネルゾーン内で SSM を有効にするには、`host-compatible=adi` 修飾子を設定できます。あとでカーネルゾーンを SSM が使用可能でない古い SPARC ベースサーバーや以前のバージョンの Oracle Solaris に移行する場合は、このプロパティを互換性のある値に設定するか、それをクリアしてゾーンがターゲットシステムの環境で機能できるようにする必要があります。

`host-compatible` 修飾子は、リリース固有のほかの機能を有効にするためにも使用できます。詳細は、『[Oracle Solaris ゾーン構成リソース](#)』の「[solaris-kz SPARC のみ: カーネルゾーンの移行クラスとホスト互換性レベル](#)」を参照してください。

例 19 カーネルゾーンでの SSM の有効化

この例では、`host-compatible` が設定されていないことを示してから、それを `adi` に設定し、ゾーンをブートします。

```
global# zonecfg -z kzone1
zonecfg:kzone1> info host-compatible
zonecfg:kzone1> set host-compatible=adi
zonecfg:kzone1> exit
global# zonecfg -z kzone1 boot
```

例 20 Silicon Secured Memory 機能のないシステムのカーネルゾーンで SSM の有効化を試みる

この例では、SSM をサポートしていない SPARC T5 システム上で SSM を有効にする試みを示します。カーネルゾーンをブートするまでエラーは検出されません。

```
global# zonecfg -z kzone1
zonecfg:kzone1> set host-compatible=adi
zonecfg:kzone1> exit
```

```
global# zonecfg -z kzone1 boot
zone 'kzone1': error: modifier adi not supported by migration class SPARC-T5
```

例 21 以前のシステムに移行できるように `host-compatible` プロパティをクリアする

この例では、SSM などの機能をサポートしていないターゲットホストにカーネルゾーンを移行できるように、`host-compatible` プロパティをクリアする方法を示します。クリアしたあとでリブートする必要があります。

```
global# zonecfg -z kzone1 clear host-compatible
zonecfg:kzone1> zoneadm -z kzone1 reboot
```


Oracle Solaris カーネルゾーンのインストール、停止処理、およびクローニング

この章では、複数の方法を使用してカーネルゾーンをインストールする方法、カーネルゾーンをアンインストールする方法、およびカーネルゾーンを停止、停止処理、再起動、およびクローニングする方法について説明します。この章には次のトピックが含まれます。

- 41 ページの「カーネルゾーンのインストール」
- 49 ページの「カーネルゾーンのアンインストール」
- 49 ページの「カーネルゾーンの停止処理、リブート、および停止」
- 50 ページの「カーネルゾーンのクローニング」

ゾーンのインストールおよびゾーンのクローニングの概念に関する全般的な情報については、『[Oracle Solaris ゾーンを紹介](#)』を参照してください。

カーネルゾーンのインストール

カーネルゾーンをインストールする前に、[第1章「Oracle Solaris カーネルゾーンの計画と構成」](#)の説明に従って構成する必要があります。構成後に、カーネルゾーンは `zoneadm install` コマンドを使用してインストールします。

カーネルゾーンは、次のいずれかの方法でインストールできます。

- カーネルゾーンの直接インストール。[42 ページの「直接インストールを使用したカーネルゾーンのインストール」](#)を参照してください。
- Automated Installation (AI) マニフェストまたは Oracle Solaris システム構成 (`sysconfig`) プロファイル。[44 ページの「カーネルゾーンインストールでの AI マニフェストと `sysconfig` プロファイルの使用」](#)を参照してください。
- Oracle Solaris インストールイメージ。[48 ページの「インストールイメージからのカーネルゾーンのインストール」](#)を参照してください。

すでにインストールされているカーネルゾーンをクローニングすることもできます。[50 ページの「カーネルゾーンのクローニング」](#)を参照してください。

直接インストールを使用したカーネルゾーンのインストール

直接インストールは、カーネルゾーンのデフォルトのインストール方法です。直接インストールでは、インストーラは大域ゾーンで実行されます。デフォルトでは、インストーラは大域ゾーンの pkg パブリッシャーを使用して、カーネルゾーンのブートディスクを作成およびフォーマットし、Oracle Solaris パッケージをそのディスクにインストールします。

注記 - カーネルゾーンの直接インストールでは、インストーラは、大域ゾーンで実行中の同じバージョンの Oracle Solaris のみを認識してインストールできます。大域ゾーンにインストールされているバージョンとは異なる Oracle Solaris バージョンをインストールするには、自動インストールまたは対話型テキストインストールを使用する必要があります。[48 ページの「インストールイメージからのカーネルゾーンのインストール」](#)を参照してください。

カーネルゾーンの直接インストールは、zoneadm install 操作中に -b オプションを指定しないと実行されます。

▼ 直接インストールを使用してカーネルゾーンをインストールする方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. カーネルゾーンをインストールします。

```
# zoneadm -z zonename install
```

たとえば、ホストシステム global にカーネルゾーン kzone1 をインストールするには:

```
global# zoneadm -z kzone1 install
```

注記 - ゾーンの検証後に直接インストールが失敗する場合は、必要なすべてのパッケージコンポーネントが大域ゾーンのパブリッシャーにあることを確認します。詳細は、『[Oracle Solaris 11.3 パッケージリポジトリのコピーと作成](#)』を参照してください。

3. カーネルゾーンをブートします。

```
# zoneadm -z zonename boot
```

たとえば、ホストシステム `global` でカーネルゾーン `kzone1` をブートするには:

```
global# zoneadm -z kzone1 boot
```

4. カーネルゾーンのコンソールにログインし、ゾーンの構成プロセスを完了します。

```
# zlogin -C zonename
```

たとえば、カーネルゾーン `kzone1` でコンソールにログインするには、次のようになります。

```
global# zlogin -C kzone1
```

例 22 直接インストールを使用したカーネルゾーンのインストール

この例は、カーネルゾーン `kzone1` の直接インストールが成功したことを示しています。

```
global# zoneadm -z kzone1 install
Progress being logged to /var/log/zones/zoneadm.20146T195713Z.kzone1.install
pkg cache: Using /var/pkg/publisher.
Install Log: /system/volatile/install.778521/install_log
AI Manifest: /tmp/zoneadm777933.spq5FV/devel-ai-manifest.xml
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Installation: Starting ...

    Creating IPS image
    Startup: Retrieving catalog 'nightly' ... Done
    Startup: Caching catalogs ... Done
    Startup: Refreshing catalog 'nightly' ... Done
    Startup: Refreshing catalog 'solaris' ... Done
    Startup: Refreshing catalog 'extra' ... Done
    Startup: Caching catalogs ... Done
    Installing packages from:
        solaris
            origin: http://ipkg.us.oracle.com/solaris11/dev/
            Startup: Linked image publisher check ... Startup: Refreshing catalog
'nightly' ... Done
            Startup: Refreshing catalog 'solaris' ... Done
            Startup: Refreshing catalog 'extra' ... Done
    Planning: Solver setup ... Done
    Planning: Running solver ... Done
    Planning: Finding local manifests ... Done
    Planning: Fetching manifests: 0/477 0% complete
    Planning: Fetching manifests: 477/477 100% complete
    Planning: Package planning ... Done
    Planning: Merging actions ... Done
    Planning: Checking for conflicting actions ... Done
    Planning: Consolidating action changes ... Done
    Planning: Evaluating mediators ... Done
    Planning: Planning completed in 29.49 seconds
    The following licenses have been accepted and not displayed.
    Please review the licenses for the following packages post-install:
        consolidation/osnet/osnet-incorporation
    Package licenses may be viewed using the command:
        pkg info --license <pkg_fmri>

    Download:      0/52325 items      0.0/535.0MB  0% complete
    Download:    1024/52325 items     30.8/535.0MB  5% complete
    Download:    2233/52325 items     42.7/535.0MB  7% complete
```

```
...
Download: 46744/52325 items 518.8/535.0MB 96% complete (6.4M/s)
Download: Completed 534.98 MB in 79.80 seconds (5.0M/s)
Actions: 1/74042 actions (Installing new actions)
Actions: 17036/74042 actions (Installing new actions)
...
Actions: 72796/74042 actions (Installing new actions)
Actions: Completed 74042 actions in 97.96 seconds.
Done
Installation: Succeeded
Done: Installation completed in 359.901 seconds.
```

カーネルゾーンインストールでの AI マニフェストと sysconfig プロファイルの使用

大域ゾーンとは別に、特定のリソースおよびパッケージ構成で複数のカーネルゾーンをインストールする必要がある場合は、Automated Installation (AI) マニフェストまたは sysconfig プロファイルを使用できます。

代替 AI マニフェストを指定するには、`zoneadm install` コマンドに `-m` オプションを付けて使用します。

```
# zoneadm -z zonename install -m manifest
```

sysconfig プロファイルを指定するには、`-c` オプションを使用します。

```
# zoneadm -z zonename install -c sysconfig-profile
```

たとえば、AI マニフェスト `/data/archives/kzone-manifest.xml` を使用してカーネルゾーン `kzone1` をインストールするには、次のようにします。

```
global# zoneadm -z kzone1 install -m /data/archives/kzone-manifest.xml
```

代替 AI マニフェストまたは sysconfig プロファイルをカーネルゾーンにインストールする場合のガイドラインは次のとおりです。

- インストールを成功させるには、AI マニフェストおよび sysconfig ファイルにフルパスおよび `.xml` 接尾辞を含める必要があります。
- AI マニフェストのカスタムの `disk` 参照はカーネルゾーンのインストールに適用できません。カーネルゾーンのルートディスクは大域ゾーンでは使用できないため、ルートディスクを作成できるように、カーネルゾーンのインストールスクリプトによって、ラベル付きのループバックファイル (`lofi`) デバイスが構成中に自動的に割り当てられます。カーネルゾーンで CD-ROM デバイスとして機能する、リムーバブルループバックファイル `lofi` デバイスを構成できます。67 ページの「[カーネルゾーンでのリムーバブルデバイスの管理](#)」を参照してください。

AI マニフェストの開発およびカスタマイズの詳細は、『[Oracle Solaris 11.3 システムのインストール](#)』の第9章、「[AI クライアントへのカスタマイズの割り当て](#)」を参照してください。ゾーンのルートディスク作成の詳細は、『[Oracle Solaris](#)」

『ゾーンの作成と使用』の「ゾーンのインストールと管理の概要」を参照してください。

- 大域ゾーンにインストールされているものとは異なる Oracle Solaris バージョンをインストールするために AI マニフェストを使用する場合は、インストールする Oracle Solaris バージョン用のイメージからインストールを実行する必要があります。例については、48 ページの「インストールイメージからのカーネルゾーンのインストール」を参照してください。

例 23 個別の Automated Installer (AI) マニフェストを使用したカーネルゾーンのインストール

この例は、デフォルト以外の Automated Install (AI) マニフェスト `/var/tmp/kz_manifest.xml` を使用したカーネルゾーン `kzone1` のインストールを示しています。

```
global# zoneadm -z kzone1 install -m /var/tmp/kz_manifest.xml
Progress being logged to /var/log/zones/zoneadm.20146T195713Z.kzone1.install
pkg cache: Using /var/pkg/publisher.
  Install Log: /system/volatile/install.10708/install_log
  AI Manifest: /tmp/zoneadm10343.5la4Vu/devel-ai-manifest.xml
  SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Installation: Starting ...

  Creating IPS image
  Startup: Retrieving catalog 'solaris' ... Done
  Startup: Caching catalogs ... Done
  Startup: Refreshing catalog 'solaris' ... Done
  Installing packages from:
  solaris
    origin: http://pkg.oracle.com/solaris/release/
  Startup: Linked image publisher check ... Startup: Refreshing catalog
'solaris' ... Done
  Planning: Solver setup ... Done
  Planning: Running solver ... Done
  Planning: Finding local manifests ... Done
  Planning: Fetching manifests: 0/501 0% complete
  Planning: Fetching manifests: 501/501 100% complete
  Planning: Package planning ... Done
  Planning: Merging actions ... Done
  Planning: Checking for conflicting actions ... Done
  Planning: Consolidating action changes ... Done
  Planning: Evaluating mediators ... Done
  Planning: Planning completed in 32.07 seconds
  The following licenses have been accepted and not displayed.
  Please review the licenses for the following packages post-install:
    consolidation/osnet/osnet-incorporation
  Package licenses may be viewed using the command:
    pkg info --license <pkg_fmri>

  Download: 0/64687 items 0.0/569.3MB 0% complete
  Download: 931/64687 items 5.8/569.3MB 1% complete (1.2M/s)
  ...
  Download: 64589/64687 items 569.2/569.3MB 99% complete (825k/s)
  Download: Completed 569.25 MB in 358.54 seconds (1.6M/s)
  Actions: 1/88614 actions (Installing new actions)
  Actions: 19471/88614 actions (Installing new actions)
  ...
  Actions: 86994/88614 actions (Installing new actions)
```

```
Actions: 87128/88614 actions (Installing new actions)
Actions: Completed 88614 actions in 73.71 seconds.
Installation: Succeeded
Done: Installation completed in 342.508 seconds.
```

```
Log saved in non-global zone as /zones/kzone1/root/var/log/zones/
zoneadm.20146T195713Z.kzone1.install
global#
```

例 24 非ルートプールを含む統合アーカイブ (UAR) に対する Automated Installer (AI) マニフェストを使用したカーネルゾーンのインストール

UAR に非ルートプール内のデータセットが含まれており、AI マニフェストが非ルートプールを考慮しない場合、次のエラーが表示されることがあります。

```
ERROR: Archive contains non-root data, please use [-m manifest]
```

次のサンプル AI マニフェストは、パス `/Extpool/Archive/Clone-T4.uar` にある統合アーカイブ (UAR) からのインストール用です。このアーカイブは `tank` という名前の非ルート `zpool` があるシステムに作成されています。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE auto_install SYSTEM "file:///usr/share/install/ai.dtd.1">
<auto_install>
  <ai_instance name="default">
    <target name="origin">
      <disk in_zpool="rpool" in_vdev="rpool-none" whole_disk="true">
        <disk_name name="c1d0" name_type="ctd"/>
      </disk>
      <disk in_zpool="tank" in_vdev="tank-none" whole_disk="true">
        <disk_name name="c1d1" name_type="ctd"/>
      </disk>
      <logical noswap="false" nodump="false">
        <zpool name="rpool" action="create" is_root="true"
          mountpoint="/rpool">
          <vdev name="rpool-none" redundancy="none"/>
        </zpool>
        <zpool name="tank" action="create" is_root="false"
          mountpoint="/tank">
          <vdev name="tank-none" redundancy="none"/>
        </zpool>
      </logical>
    </target>
    <software type="ARCHIVE">
      <source>
        <file uri="file:///Extpool/Archive/Clone-T4.uar"/>
      </source>
      <software_data action="install">
        <name>*</name>
      </software_data>
    </software>
  </ai_instance>
</auto_install>
```

マニフェストファイルが `/tmp/ai.xml` に格納されており、ID 0 および 1 のストレージデバイスが `kzone1` ゾーン構成に存在している場合、次のコマンドを使用して、カーネルゾーン `kzone1` にインストールできます。

```
global# zoneadm -z kzone1 install -m /tmp/ai.xml
```

例 25 代替 sysconfig プロファイルを使用したカーネルゾーンのインストール

この例では、sysconfig プロファイル /var/tmp/kzone1-sysconfig.xml を使用してカーネルゾーン kzone1 をインストールします。

```
global# zoneadm -z kzone1 install -c /var/tmp/kzone1-sysconfig.xml
The following ZFS file system(s) have been created:
  rpool/zones/kzone1
Progress being logged to /var/log/zones/zoneadm.20146T195713Z.kzone1.install
pkg cache: Using /var/pkg/publisher.
AI Manifest: /tmp/zoneadm124827.zQwo0h/devel-ai-manifest.xml
SC Profile: /var/tmp/kzone1-sysconfig.xml
Installation: Starting ...
```

```

Creating IPS image
Startup: Retrieving catalog 'nightly' ... Done
Startup: Caching catalogs ... Done
Startup: Refreshing catalog 'nightly' ... Done
Startup: Refreshing catalog 'solaris' ... Done
Startup: Refreshing catalog 'extra' ... Done
Startup: Caching catalogs ... Done
Installing packages from:
  nightly
    origin: file:///server/nightly
  solaris
    origin: file:///server/solaris
  extra
    origin: http://server/extra
Startup: Refreshing catalog 'nightly' ... Done
Startup: Refreshing catalog 'solaris' ... Done
Startup: Refreshing catalog 'extra' ... Done
Planning: Solver setup ... Done
Planning: Running solver ... Done
Planning: Finding local manifests ... Done
...
Planning: Fetching manifests: 552/552 100% complete
Planning: Package planning ... Done
Planning: Merging actions ... Done
Planning: Checking for conflicting actions ... Done
Planning: Consolidating action changes ... Done
Planning: Evaluating mediators ... Done
Planning: Planning completed in 56.62 seconds
...
Download: 9746/65597 items 143.6/661.7MB 21% complete
Download: 35018/65597 items 370.8/661.7MB 56% complete
Download: 62181/65597 items 654.5/661.7MB 98% complete
Download: Completed 661.67 MB in 40.57 seconds (0B/s)
...
Actions: 87940/89672 actions (Installing new actions)
Actions: 88107/89672 actions (Installing new actions)
Actions: 88745/89672 actions (Installing new actions)
Actions: Completed 89672 actions in 108.50 seconds.
Done
Installation: Succeeded
      Done: Installation completed in 342.508 seconds.
```

```
Log saved in non-global zone as /zones/kzone1/root/var/log/zones/
zoneadm.20146T195713Z.kzone1.install
global#
```

インストールイメージからのカーネルゾーンのインストール

-b オプションを使用することで、Oracle Solaris ISO インストールイメージからカーネルゾーンをインストールできます。

```
# zoneadm -z zonename install -b path-to-iso-file
```

次の点に注意してください。

- 対話型テキストインストールとメディアからの自動インストールの両方がサポートされています。ライブメディアインストールはカーネルゾーンでサポートされていません。これらのインストール方法の詳細は、『Oracle Solaris 11.3 システムのインストール』のパート 2、「インストールメディアを使用したインストール」を参照してください。
- インストールイメージ内の Oracle Solaris のバージョンはカーネルゾーンをサポートしている必要があるため、Oracle Solaris 11.2 以上である必要があります。15 ページの「カーネルゾーンホストでのハードウェアおよびソフトウェアサポートの確認」を参照してください。
- インストールを開始するには、ISO イメージへの完全なパスを含める必要があります。そうでない場合、Oracle Solaris のインストールは失敗します。
- 特定のリソースおよびパッケージ構成を含む AI マニフェストと組み合わせたインストールイメージからインストールできます。

ISO ファイルからの Oracle Solaris のインストール中に、カーネルゾーンがブートされ、ゾーンコンソールに接続されます。ゾーンコンソールの使用方法については、『Oracle Solaris ゾーンの作成と使用』の「ゾーンコンソールログイン」を参照してください。



注意 - インストールが完了する前に、カーネルゾーンコンソールを終了するか、切断すると、インストールが失敗します。

▼ インストールイメージからカーネルゾーンをインストールする方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. Oracle Solaris インストールイメージを使用してカーネルゾーンをインストールします。


```
global# zoneadm -z zonename install -b path-to-iso-file
```

たとえば、`/var/tmp/solaris-media.iso`にあるイメージをカーネルゾーン `kzone2` にインストールするには:

```
global# zoneadm -z kzone2 install -b /var/tmp/solaris-media.iso
```

▼ インストールイメージからカーネルゾーンをインストールし、AI マニフェストを使用する方法

1. 管理者になります。

詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. Oracle Solaris インストールイメージを使用してカーネルゾーンをインストールし、`-m` オプションを使ってマニフェストファイルを指定します。

`/var/tmp/solaris-media.iso`にあるイメージをカーネルゾーン `kzone2` にインストールし、特定のリソースおよびパッケージ構成を含む AI マニフェスト `/var/tmp/kz_manifest.xml` も使用するには:

```
global# zoneadm -z kzone2 install -b /var/tmp/solaris-media.iso -m /var/tmp/kz_manifest.xml
```

カーネルゾーンのアンインストール

たとえば新規または更新済みのゾーン構成をインストールする前にカーネルゾーンをアンインストールするには、`zoneadm uninstall` コマンドを使用します。この操作の実行中は、ゾーンを実行状態にはできません。ゾーンのアンインストール手順については、『[Oracle Solaris ゾーンの作成と使用](#)』の「[ゾーンの停止処理、停止、リブート、およびアンインストール](#)」を参照してください。

ゾーンをアンインストールするには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

カーネルゾーンの停止処理、リブート、および停止

カーネルゾーンを停止処理、リブート、および停止するには、`zoneadm shutdown`、`zoneadm reboot`、および `zoneadm halt` コマンドを使用します。これらのコマンドの使用については、『[Oracle Solaris ゾーンの作成と使用](#)』の「[ゾーンの停止処理、停止、リブート、およびアンインストールについて](#)」を参照してください。

ホストシステムがリブートするときにゾーンを自動的にブートするには、`autoboot zonecfg` リソースを設定します。このリソースの設定方法の詳細は、『[Oracle Solaris ゾーンの作成と使用](#)』の第1章、「[非大域ゾーンの計画および構成方法](#)」を参照してください。

ゾーンを停止処理、リブート、および停止するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

カーネルゾーンのクローニング

クローニングにより、システムの既存の構成済みおよびインストール済みゾーンを同一のシステム上の新しいゾーンにコピーできます。クローニングされたゾーンには、既存のゾーンのカスタマイズが含まれます。たとえば、追加されたパッケージ、変更されたゾーンリソース、ソースゾーンでのファイルの変更が、クローニングされた各ゾーンに表示されます。ゾーンのクローニングは、カスタマイズされた同様のゾーン構成でさらにゾーンを追加する場合に効率的な方法です。

カーネルゾーンは次の方法でクローニングできます。

- 少数のゾーンをクローニングする必要がある場合、`zoneadm clone` コマンドを使用します。例26「[zoneadm clone コマンドを使用したカーネルゾーンのクローニング](#)。」を参照してください。
- データセンター環境などの大規模な配備で複数のゾーンをクローニングする必要がある場合は、統合アーカイブファイルを使用します。例27「[統合アーカイブを使用したカーネルゾーンのクローニングと配備](#)」を参照してください。

注記 - 統合アーカイブファイルには、実行中の状態のカーネルゾーンのみを含めることができます。統合アーカイブの作成中に、実行中ではないカーネルゾーンを除外できます。詳細は、『[Oracle Solaris 11.3 でのシステム復旧とクローン](#)』の第2章、「[統合アーカイブの操作](#)」を参照してください。

カーネルゾーンのクローニング後に、新しいゾーンをブートしてログインできます。

例 26 `zoneadm clone` コマンドを使用したカーネルゾーンのクローニング。

この例は、カーネルゾーン `kzone1` をホストシステム `global` のカーネルゾーン `kzone2` にクローニングする方法を示しています。手順については、『[Oracle Solaris ゾーンの作成と使用](#)』の「[同一システム上での非大域ゾーンのクローニング](#)」を参照してください。

```
global# zoneadm -z kzone1 halt
```

```

global# zonecfg -z kzone2 create -t kzone1
global# zoneadm -z kzone2 clone kzone1
Progress being logged to /var/log/zones/zoneadm.20140327T223951Z.kzone2.clone
Install Log: /system/volatile/install.100847/install_log
  AI Manifest: /system/shared/ai.xml
Installation: Starting ...

    Creating direct clone image...
    Registering dynamic archive transfer
    Pre-validating manifest targets before actual target selection
    Pre-validation of manifest targets completed
    Validating combined manifest and archive origin targets
    Commencing transfer of stream: ...
    Completed transfer of direct stream: ...
    Archive transfer completed
Installation: Succeeded

```

例 27 統合アーカイブを使用したカーネルゾーンのクローニングと配備

次の例は、`archiveadm` コマンドを使用したカーネルゾーン `kzone1` のクローニングと配備を示しています。カーネルゾーン `kzone1` 用の統合アーカイブが作成されます。アーカイブ情報が検証され、`kzone1` から変更されたゾーン構成でカーネルゾーン `kzone2` がクローニングされます。手順については、『[Oracle Solaris 11.3 でのシステム復旧とクローン](#)』を参照してください。

```

global# archiveadm create -z kzone1 /var/tmp/kzone1.uar
Unified Archive initialized: /var/tmp/kzone1.uar.
\
Logging to: /system/volatile/archive_log.26248
Dataset discovery completed...
/
Media creation complete for zone(s)...
-
Archive stream creation completed...
-
Archive creation completed...
global# zoneadm list -cv
  ID NAME          STATUS    PATH                                BRAND  IP
  0 global          running   /                                    solaris shared
  2 kzone1          running   -                                    solaris-kz excl
global# archiveadm info /var/tmp/kzone1.uar
Archive Information
  Creation Time: 2014-04-10T17:12:12Z
  Source Host: global
  Architecture: i386
  Operating System: Oracle Solaris 11.2 X86
  Deployable Systems: kzone1
global# zonecfg -z kzone2 create -a /var/tmp/kzone1.uar
global# zoneadm -z kzone2 install -a /var/tmp/kzone1.uar
global# zoneadm list -cv
  ID NAME          STATUS    PATH                                BRAND  IP
  0 global          running   /                                    solaris shared
  2 kzone1          running   -                                    solaris-kz excl
  - kzone2          configured -                                    solaris-kz excl

```


Oracle Solaris カーネルゾーンの移行

ゾーンの移行は、既存のゾーンまたはシステムを別のシステム上のゾーンに転送します。この章では、カーネルゾーンの移行の方法と管理について説明します。

カーネルゾーンの移行の要件

カーネルゾーンのゾーン移行の要件は、ほかのブランドゾーンと同じです。これらの要件の詳細は、『[Oracle Solaris ゾーンの作成と使用](#)』の第 7 章、「[Oracle Solaris ゾーンの移行および変換](#)」で説明されています。

注記 - SPARC と x86 のアーキテクチャー間のゾーン移行はサポートされていません。

さらに、ソースホストとターゲットホストの両方が、Oracle Solaris カーネルゾーンをサポートしている必要があります。[13 ページの「Oracle Solaris カーネルゾーンのハードウェアおよびソフトウェア要件」](#)を参照してください。

注記 - カーネルゾーンのライブ移行には、特定のシステムファームウェアおよびストレージ要件があります。詳細は、[56 ページの「ライブ移行の要件」](#)を参照してください。

コールド移行を使用したカーネルゾーンの移行

コールド移行を使用して、カーネルゾーンを移行できます。コールド移行では、ゾーンが停止されて移動され、別のホスト上でリブートされます。コールド移行は、時間が重要となるサービスを提供するアプリケーションまたはメモリーフットプリントが大きいアプリケーションの移行に使用されます。

コールド移行は、すべてのゾーンブランドでサポートされています。コールド移行の詳細については、『[Oracle Solaris ゾーンの作成と使用](#)』の第 7 章、「[Oracle Solaris ゾーンの移行および変換](#)」を参照してください。

カーネルゾーンのクールド移行の一環として、宛先ホストの暗号化鍵を定義する必要があります。63 ページの「暗号化鍵とカーネルゾーンの移行」を参照してください。

ウォーム移行を使用したカーネルゾーンの移行

zoneadm suspend コマンドを使用してゾーンをその現在のホストで一時停止し、新しいホストで再開することで、カーネルゾーンを別のホストに移行できます。このゾーン移行方式は、ウォーム移行または一時停止および再開を使用した移行と呼ばれます。

ウォーム移行では、カーネルゾーンの実行中にシステムを完全にリブートしてアプリケーションを再起動する必要はありません。

ウォーム移行では、ソースホストとターゲットホストの両方で互換性を持つようにゾーン構成を設定する必要があります。ゾーン構成の非互換性およびウォーム移行の詳細については、[solaris-kz\(5\)](#) マニュアルページを参照してください。

ウォーム移行では、ゾーンの suspend リソースが、ソースホストとターゲットホストの両方からアクセスできる共有ストレージ用に構成されている必要があります。34 ページの「suspend リソースの構成」および『Oracle Solaris ゾーンの作成と使用』の第 13 章、「共有ストレージでの Oracle Solaris ゾーンの使用開始」を参照してください。

カーネルゾーンのウォーム移行の一環として、宛先ホストの暗号化鍵を定義する必要があります。63 ページの「暗号化鍵とカーネルゾーンの移行」を参照してください。

▼ ウォーム移行を使用してカーネルゾーンを移行する方法

始める前に [34 ページの「suspend リソースの構成」](#) を参照してください。

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. 移行するカーネルゾーンの suspend リソースに共有ストレージが構成されていることを確認します。
例:

```
global# zonecfg -z kzone1 info suspend
suspend:
    storage: iscsi://system/luname.naa.501337600144f0dbf8af1900
```

3. 大域ゾーンで、移行するカーネルゾーンのファイルシステムを一時停止します。

```
# zoneadm -z zonename suspend
```

たとえば、大域ゾーン `global` でカーネルゾーン `kzone1` を一時停止するには、次のようにします。

```
global# zoneadm -z kzone1 suspend
```

4. 大域ゾーンのカーネルゾーンのファイルシステムを切り離します。

```
# zoneadm -z zonename detach
```

たとえば、大域ゾーン `global` でカーネルゾーン `kzone1` を切り離すには、次のようにします。

```
global# zoneadm -z kzone1 detach
```

5. ゾーン構成をエクスポートし、ファイルをターゲットホストに転送します。

また、この手順では、ソースと同じ構成を使用してターゲットホストシステム上のゾーンを構成します。

```
# zonecfg -z zonename export | ssh root@newhost zonecfg -z zonename -f -
```

例:

```
global# zonecfg -z kzone1 export | ssh root@global2 zonecfg -z kzone1 -f -
```

6. 新規ホスト上でゾーンを接続します。

```
# zoneadm -z zonename attach
```

例:

```
global2# zoneadm -z kzone1 attach
```

7. 新しいホストでカーネルゾーンをブートし、移行したゾーンを再開します。

```
# zoneadm -z zonename boot
```

例:

```
global2# zoneadm -z kzone1 boot
```

ライブ移行を使用したカーネルゾーンの移行

ライブ移行を使用すると、`running` 状態のカーネルゾーンを新しいカーネルゾーンホストに移行できます。カーネルゾーンのメモリーの状態が移行されたゲストにコピー

されるため、ライブ移行で発生する停止時間は、ほとんどのアプリケーションまたはほとんどのエンドユーザーが気づかないほど短くなります。

アプリケーションの停止時間を最小限に抑える必要がある場合や、アプリケーションを `running` 状態のままにする必要がある場合に、ライブ移行を使用できます。

ライブ移行の要件

ソースホストとターゲットホストの両方がカーネルゾーンのハードウェアおよびソフトウェアの最小要件 ([13 ページの「Oracle Solaris カーネルゾーンのハードウェアおよびソフトウェア要件」](#)を参照) を満たすことに加え、ソースシステムとターゲットシステムの両方が次のライブ移行の要件を満たしている必要があります。

- 両方のホストのオペレーティングシステムが Oracle Solaris 11.3 である必要があります。
- SPARC ベースシステム間で移行する場合は、次のファームウェアバージョンがインストールされている必要があります。
 - システムファームウェア 8.8 以上がインストールされている SPARC T4 システム
 - システムファームウェア 9.5 以上がインストールされている SPARC T5、SPARC M5、または SPARC M6 システム
 - SPARC T7 または SPARC M7 シリーズサーバー。すべてのファームウェアバージョンがサポートされています。
 - XCP ファームウェア 2280 以上がインストールされている富士通 M10 または SPARC M10 サーバー。
- ファームウェアバージョン要件に加え、異なる SPARC アーキテクチャー間で移行する場合は、[63 ページの「SPARC カーネルゾーンのウォーム移行およびライブ移行用の CPU 移行クラスの指定」](#)の説明に従って、`cpu-arch` プロパティを設定する必要があります。

さらに、ライブ移行のソースホストとターゲットホストには次が必要です。

- ゾーンによって使用されるすべてのストレージは、ソースホストとターゲットホストの両方からアクセスできる共有ストレージを持っている必要があります。『[Oracle Solaris ゾーンの作成と使用](#)』の第 13 章、「[共有ストレージでの Oracle Solaris ゾーンの使用開始](#)」を参照してください。ローカル専用ディスクパスはライブ移行ではサポートされません。
- ソースホストとターゲットホストの両方に互換性を持ち整合性を保つようにゾーン構成を設定する必要があります。『[Oracle Solaris ゾーン構成リソース](#)』を参照してください。
- 次のサービスの実行中のインスタンス。
 - カーネルゾーンのライブ移行サービス、`svc:/network/kz-migr:stream`。ターゲットホストのポート 8102 が開いている必要があります。

- Oracle Solaris リモート管理デーモン (RAD)。TCP、TLS、および SSH トランスポートがサポートされています。RAD の詳細は、『[Remote Administration Daemon Developer's Guide](#)』を参照してください。
- NTP (Network Time Protocol) サーバー。NTP の詳細は、『[Oracle Solaris 11 ネットワークサービスの紹介](#)』の「[ネットワークキャッシングおよび時間関連のサービスを管理するための主要なタスク](#)」を参照してください。

Oracle Solaris サービスの管理の詳細は、『[Oracle Solaris 11.3 でのシステムサービスの管理](#)』の第 3 章、「[サービスの管理](#)」を参照してください

- ソースホストとターゲットホストの間で構成済みの SSH 公開鍵認証。公開鍵認証は、SSH がプロンプトを必要としないように構成されている必要があります。『[Oracle Solaris 11.3 での Secure Shell アクセスの管理](#)』の「[Secure Shell で使用する公開鍵と非公開鍵のペアを生成する方法](#)」を参照してください。

カーネルゾーンのライブ移行には、10 Gb Ethernet リンクを使用することをお勧めします。

root 以外のユーザーによるカーネルゾーンのライブ移行操作の実行の許可

Zone Migration 権利プロファイルは、root 以外のユーザーが `zoneadm migrate` を使用してライブ移行によってカーネルゾーンを移行できるようにします。

次の出力に、Zone Migration 権利プロファイルを持つユーザー `kz-user1` の承認が表示されます。

```
$ auths kz-user1
solaris.admin.wusb.read,solaris.mail.mailq,solaris.network.autoconf.read,solaris.zone.migrate/vz1-112
$ profiles kz-user1
Zone Migration
Basic Solaris User
All
```

ユーザーは、移行されるゾーンの構成を作成するために、ターゲットシステムにも Zone Configuration 権利プロファイルが必要です。

Oracle Solaris で権利プロファイルの割り当てと管理を行う方法については、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護](#)』を参照してください。

zoneadm migrate コマンドについて

ライブ移行を実行するには、`zoneadm migrate` コマンドを使用します。`zoneadm migrate` コマンドオプションには次のものがあります。

- n ライブ移行の予行演習を行います。ゾーンをターゲットホストにライブ移行できるかどうかを確認します。
ゾーンは `running` 状態です。
- q ライブ移行の非出力モード。ライブ移行操作中にステータスを報告しないように指定します。
- c *cipher* ライブ移行のセキュアな暗号化オプションを指定します。デフォルトで、特定の暗号化を指定しない場合でも、両方のシステムでサポートされている暗号化を使用して、移行がセキュリティー保護されます。62 ページの「[セキュアなライブ移行について](#)」を参照してください。
- ssh|rads:
//user@host:port ゾーンをターゲットホストに移行するために使用するスキーム、ユーザー名、およびホスト名を含む RAD URI を指定します。ssh スキームは SSH を使用し、rads スキームは TLS を使用します。ホスト名のみを指定した場合、スキームのデフォルト値は `rads` になり、`user` のデフォルト値は現在のユーザーになり、`port` のデフォルト値は標準 RAD ポート 12302 になります。詳細については、『[Remote Administration Daemon Developer's Guide](#)』の「[Connecting to a RAD Instance by Using an URI](#)」を参照してください。
- migrate コマンドに関する詳細については、[zoneadm\(1M\)](#) のマニュアルページを参照してください。

▼ ライブ移行を使用してカーネルゾーンを移行する方法

- 始める前に カーネルゾーンのソースホストとターゲットホストの両方が、56 ページの「[ライブ移行の要件](#)」で説明されているライブ移行のためのハードウェア、ソフトウェア、およびストレージ要件を満たしていることを確認します。
1. **ライブ移行の管理権利を取得します。**
詳細については、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」または 57 ページの「[root 以外のユーザーによるカーネルゾーンのライブ移行操作の実行の許可](#)」を参照してください。
 2. **ソースホストとターゲットホストの間の対話型プロンプトを必要としないように SSH 認証を構成します。**
『[Oracle Solaris 11.3 での Secure Shell アクセスの管理](#)』の「[Secure Shell で使用する公開鍵と非公開鍵のペアを生成する方法](#)」を参照してください。

3. ターゲットホスト上で `date` などのコマンドを実行することによって SSH 認証をテストします。

```
global1# ssh global2 date
Mon Mar  9 13:22:40 PDT 2015
```

パスワードの入力を求められた場合は、対話型認証のないログインを有効にするように鍵ペアを構成していません。

『Oracle Solaris 11.3 での Secure Shell アクセスの管理』の「Secure Shell で使用する公開鍵と非公開鍵のペアを生成する方法」を参照してください。

4. ソースホストとターゲットホストで、カーネルゾーンの移行サービスを開始します。

```
global1# svcadm enable -rs svc:/network/kz-migr:stream
global1# ssh global2 svcadm enable -rs svc:/network/kz-migr:stream
```

5. ソースホストとターゲットホストで、RAD サービスのステータスをチェックします。

必要なサービスは、使用する予定の RAD URI によって異なります。ssh を使用して移行する場合、`svc:/system/rad:local` サービスが実行している必要があります。rads を使用する場合、`svc:/system/rad:remote` サービスが実行している必要があります。

```
global1# svcs rad
STATE      STIME      FMRI
disabled   14:47:31  svc:/system/rad:remote
online     17:28:50  svc:/system/rad:local
online     17:28:56  svc:/system/rad:local-http
```

```
global1# ssh global2 svcs rad
STATE      STIME      FMRI
disabled   Jul_01    svc:/system/rad:remote
online     Jul_01    svc:/system/rad:local-http
online     Jul_01    svc:/system/rad:local
```

6. (オプション) ソースおよびターゲットホストで必要な、無効にされている RAD サービスをすべて起動します。

たとえば、`svc:/system/rad:remote` サービスを有効にするには:

```
global1# svcadm enable -rs svc:/system/rad:remote
global1# ssh global2 svcadm enable -rs svc:/system/rad:remote
```

7. ソースホストとターゲットホストで、NTP サービスのステータスをチェックします。

```
global1# svcs ntp
STATE      STIME      FMRI
online     11:09:40  svc:/network/ntp:default
```

```
global1# ssh global2 svcs ntp
STATE      STIME      FMRI
online     11:09:45  svc:/network/ntp:default
```

NTP サービスがオンラインでない場合、その設定方法については、『Oracle Solaris 11 ネットワークサービスの紹介』の「ネットワークキャッシングおよび時間関連のサービスを管理するための主要なタスク」を参照してください。

8. ソースホストで、移行するゾーンが **running** 状態であることを確認します。

```
global1# zoneadm list -cv
ID NAME          STATUS  PATH  BRAND      IP
0  global         running /      solaris    shared
1  kzone1         running -      solaris-kz excl
-  kzone2         installed -      solaris-kz excl
```

9. ライブ移行の予行演習またはプリフライトチェックを開始します。

ソースホスト上で、`zoneadm migrate -n` を使用して、ライブ移行の前にカーネルゾーン構成をテストします。

```
global1# zoneadm -z zonename migrate -n ssh://target-host
```

たとえば、ソースホスト `global1` とターゲットホスト `global2` を使用する場合:

```
root@global1:~# zoneadm -z kzone1 migrate -n ssh://global2
zoneadm: zone 'kzone1': Importing zone configuration.
zoneadm: zone 'kzone1': Attaching zone.
zoneadm: zone 'kzone1': Booting zone in 'migrating-in' mode.
zoneadm: zone 'kzone1': Checking migration compatibility.
zoneadm: zone 'kzone1': Cleaning up.
zoneadm: zone 'kzone1': Dry-run migration successful.
```

10. カーネルゾーンを移行します。

`zoneadm migrate` を使用してカーネルゾーンを移行します。

```
# zoneadm -z zonename migrate target
```

たとえば、ソースホスト `global1` からターゲットホスト `global2` に `kzone1` を移行するには:

```
root@global1:~# zoneadm -z kzone1 migrate ssh://global2
zoneadm: zone 'kzone1': Importing zone configuration.
zoneadm: zone 'kzone1': Attaching zone.
zoneadm: zone 'kzone1': Booting zone in 'migrating-in' mode.
zoneadm: zone 'kzone1': Checking migration compatibility.
zoneadm: zone 'kzone1': Starting migration.
zoneadm: zone 'kzone1': Suspending zone on source host.
zoneadm: zone 'kzone1': Waiting for migration to complete.
zoneadm: zone 'kzone1': Halting and detaching zone on source host.
zoneadm: zone 'kzone1': Migration successful.
```

11. ゾーンが移行されており、現在ターゲットホスト上で実行していることを確認します。

```
root@global2:~# zoneadm list -cv
ID NAME          STATUS  PATH  BRAND      IP
0  global         running /      solaris    shared
1754 kzone1         running -      solaris-kz excl
-  kzone2         configured -      solaris-kz excl
-  kzone3         installed -      solaris-kz excl
-  kzone4         configured -      solaris-kz excl
```

12. (オプション) ソースホストで、移行されたゾーンが **configured** 状態であることを確認します。

```
global1# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	kzone1	configured	-	solaris-kz	excl
-	kzone2	installed	-	solaris-kz	excl

注記 - ライブ移行前に作成されたソースホストをあとでブート環境 (BE) にブートした場合、移行されたカーネルゾーンの状態が `configured` ではなく `unavailable` として表示されることがあります (ゾーンの状態は BE 間で共有されないため)。次のコマンドを発行し、ゾーンのスレージを切り離して、ゾーンの状態を `configured` に戻します。

```
# zoneadm -z zonename detach -F
```

例 28 ライブ移行のプリフライトチェックの失敗

この例は、ソースホスト `global1` とターゲットホスト `global2` の間で失敗したプリフライトチェックを示しています。 `virtual-cpu` リソースが両方のホストの間で矛盾しています。ゾーン構成についての詳細は、『[Oracle Solaris ゾーン構成リソース](#)』を参照してください。

```
global1# zoneadm -z kzone1 migrate -n ssh://global2
zoneadm: zone 'kzone1': Using existing zone configuration on destination.
zoneadm: zone 'kzone1': Attaching zone.
zoneadm: zone 'kzone1': Booting zone in 'migrating-in' mode.
zoneadm: zone 'kzone1': boot failed:
zone 'kzone1': error: Suspended zone has 8 active VCPUs, more than the configured
zone 'kzone1': virtual-cpu maximum of 4.
zone 'kzone1': error: Correct errors, or delete the configuration, using zonecfg(1M) on the
zone 'kzone1': destination host.
zoneadm: zone kzone1: call to zoneadmd(1M) failed: zoneadmd(1M) returned an error 9 (zone
state change failed)
```

例 29 2つの異なる anet 構成を持つホスト間のライブ移行

次の例では、さまざまな `anet` 構成のホスト間のライブ移行の準備を示しています。 `global1` 上のゾーン `kzone1` の構成は、ターゲットホスト `global2` で適していません。 `global2` で、ゾーンはその `anet` リソースに `net1` を使用する必要があります。構成がターゲットホストにエクスポートされ、変更されてから、プリフライトチェックが実行されます。

`anet` リソースの詳細については、『[Oracle Solaris ゾーン構成リソース](#)』を参照してください。

```
global1# zonecfg -z kzone1 -r export | ssh root@global2 zonecfg -z kzone1 -f -
global1# ssh root@global2 zonecfg -z kzone1 'select anet 0; set lower-link=net1;end'
global1# zoneadm -z kzone1 migrate -n ssh://global2
```

セキュアなライブ移行について

デフォルトでは、ライブ移行メモリー転送データはソースとターゲットホストとの間で転送されるときに、両方のホスト上でサポートされる暗号化方式を使用して暗号化されます。zoneadm migrate -c [cipher] を使用して、特定の暗号化方式を指定したり暗号化を無効化したりできます。

zoneadm migrate -c cipher には次のオプションがあります。

none	暗号化を無効にします
list	ソースおよびターゲットホストでサポートされている暗号化方式を一覧表示します。
encryption-cipher	ソースホストとターゲットホストでサポートされている暗号化方式のいずれか1つを指定します。migrate -c list コマンドによって使用可能な値が表示されます。

暗号化方式を指定しない場合は、ソースホストとターゲットホストの両方のサポートに基づいて自動的に1つ選択されます。

例 30 2つの信頼できるホスト間のライブ移行

次の例は、ソースホスト global1 から宛先ホスト global2 へのカーネルゾーン kzone1 のライブ移行を示しています。暗号化は無効になっています。

```
global1# zoneadm -z kzone1 migrate -c none root@global2
Password:
zoneadm: zone 'kzone1': Using existing zone configuration on destination.
zoneadm: zone 'kzone1': Attaching zone.
zoneadm: zone 'kzone1': Booting zone in 'migrating-in' mode.
zoneadm: zone 'kzone1': Checking migration compatibility.
zoneadm: zone 'kzone1': Starting migration.
zoneadm: zone 'kzone1': Suspending zone on source host.
zoneadm: zone 'kzone1': Waiting for migration to complete.
zoneadm: zone 'kzone1': Migration successful.
zoneadm: zone 'kzone1': Halting and detaching zone on source host.
```

例 31 ライブ移行のソースホストと宛先ホストの間で暗号化方式の互換性を確認する

次の例は、ソースホスト global1 から宛先ホスト global2 へのカーネルゾーン kzone1 のライブ移行を示しています。指定された暗号化方式 aes-128-cbc は、宛先ホストでサポートされていません。

```
global1# zoneadm -z kzone1 migrate -c aes-128-cbc ssh://global2
zoneadm: zone 'kzone1': cipher aes-128-cbc not supported by destination
zoneadm: zone 'kzone1': destination supports: aes-128-ccm aes-128-gcm
```

例 32 ライブ移行のソースホストと宛先ホストの間でサポートされている使用可能な暗号化方式を一覧表示する

次の例は、カーネルゾーン `kzone1` のライブ移行中に、サポートされている使用可能な暗号化方式を一覧表示します。ゾーンはソースホスト `global1` から宛先ホスト `global2` に移行されます。

```
global1# zoneadm -z kzone1 migrate -c list root@global2
Password:
source ciphers: aes-128-ccm aes-128-gcm none
destination ciphers: none
# echo $?
0
```

暗号化鍵とカーネルゾーンの移行

カーネルゾーンでは、ゾーンの移行後に、ターゲットホスト上に暗号化鍵が必要になります。

ライブ移行を使用してゾーンを移行する場合、ターゲットホスト上の暗号化鍵はライブ移行プロセスの一環として自動的に定義されます。

コールド移行またはウォーム移行によってゾーンを移行する場合は、ソースシステム上で `zonecfg export` コマンドを使用して、ターゲットシステム上で使用するコマンドファイルを生成します。このコマンドファイルは、カーネルゾーンの暗号化鍵の要件を満たしています。たとえば、`global1` から `global2` に移行されたゾーンのコマンドファイルを生成するには、次のようにします。

```
global1# zonecfg -z kzone1 export -f /net/.../kzone1.cfg
global2# zonecfg -z kzone1 -f /net/.../kzone1.cfg
```

SPARC カーネルゾーンのウォーム移行およびライブ移行用の CPU 移行クラスの指定

SPARC ベースシステムでのウォーム移行およびライブ移行の場合のみ、ホストシステムとは異なる CPU クラスを持つようにカーネルゾーンを構成できます。カーネルゾーンの CPU 移行クラスを指定するには、`cpu-arch zonecfg` リソースを使用します。異なる CPU アーキテクチャーに移行するための `cpu-arch zonecfg` リソースプロパティの設定に関する詳細については、『Oracle Solaris ゾーン構成リソース』の「[solaris-kz SPARC のみ: カーネルゾーンの移行クラスとホスト互換性レベル](#)」を参照してください。

カーネルゾーンの CPU 移行クラスが設定されていない場合、カーネルゾーンの CPU 移行クラスはホストと同じになります。

CPU 移行クラスの変更を有効にするには、カーネルゾーンをリブートする必要があります。

注記 - カーネルゾーンホストは常に、互換性のないプラットフォームで以前に中断されたゲストの再開を拒否します。cpu-arch クラスが互換性のない値に設定されている場合、カーネルゾーンゲストはブートしません。

例 33 SPARC ベースシステムでのカーネルゾーン移行クラスの確認および設定

次の例は、カーネルゾーン kzone1 の cpu-arch リソースを確認および設定する方法を示しています。

```
global# zonecfg -z kzone1
zonecfg:kzone1> info cpu-arch
cpu-arch: generic
zonecfg:kzone1> set cpu-arch=migration-class1
zonecfg:kzone1> info cpu-arch
cpu-arch: migration-class1
zonecfg:kzone1> exit
```

例 34 互換性のない CPU アーキテクチャーが原因でライブ移行が失敗する

この例は、SPARC T4 ホスト global1 と SPARC T5 ホスト global2 の間で試行されたライブ移行を示しています。cpu-arch プロパティは、ホスト全体で一貫性がなく、『Oracle Solaris ゾーン構成リソース』の「[solaris-kz SPARC のみ: カーネルゾーンの移行クラスとホスト互換性レベル](#)」に指示されているように、設定する必要があります。

```
global1# zoneadm -z kzone1 migrate -n ssh://global2
zoneadm: zone 'kzone1': Importing zone configuration.
zoneadm: zone 'kzone1': Attaching zone.
zoneadm: zone 'kzone1': Booting zone in 'migrating-in' mode.
zoneadm: zone 'kzone1': Checking migration compatibility.
zoneadm: zone 'kzone1': configuration check failed:
error: Suspended image cannot be resumed on current cpu migration class (SPARC-T4). Please
check cpu-arch setting in zone config or in host LDom config.
2015-05-30 22:42:59 error: request failed: failed to create VM: Operation not supported
```


◆◆◆ 第 4 章

Oracle Solaris カーネルゾーンの管理

この章では、次の Oracle Solaris カーネルゾーンの管理について説明します。

- 65 ページの「カーネルゾーン環境での作業」
- 66 ページの「カーネルゾーンと不変ゾーンの操作」
- 67 ページの「カーネルゾーンでのリムーバブルデバイスの管理」
- 69 ページの「カーネルゾーンの補助状態の操作」
- 70 ページの「入れ子にされたゾーンの管理」
- 72 ページの「カーネルゾーンのホストデータおよびホスト ID」
- 73 ページの「カーネルゾーンのブートローダーの操作」
- 75 ページの「カーネルゾーンのライブゾーン再構成」
- 75 ページの「NFS ストレージ URI とカーネルゾーン」
- 76 ページの「カーネルゾーン内のコアファイル」

solaris および solaris10 ブランドゾーンの管理の内容については、『Oracle Solaris ゾーンの実成と使用』の第 9 章、「Oracle Solaris ゾーンの実成について」を参照してください。

カーネルゾーン環境での作業

カーネルゾーン環境での作業は、大域ゾーンでの作業に非常に似ています。このセクションでは、カーネルゾーンの管理環境と大域ゾーンでの作業の主な違いについて説明します。

ゾーン内でのプロセス ID の可視性

カーネルゾーンプロセスは、カーネルゾーンホストには直接表示されません。カーネルゾーンのプロセス情報を表示するには、zlogin コマンドの後ろにプロセス管理コマ

ンドを使用する必要があります。たとえば、カーネルゾーン `kzone1` の `syslogd` のプロセス情報をカーネルゾーンホスト `global` から表示するには、次のようにします。

```
global# zlogin kzone1 ps -ef |grep syslogd
root 1520      1   0 20:23:08 ?          0:00 /usr/sbin/syslogd
```

カーネルゾーンの重複プロセス ID

大域ゾーンと各カーネルゾーンはそれぞれ独自のプロセス ID 空間を管理します。同じ数値プロセス ID で、大域ゾーンと 1 つ以上のカーネルゾーンの複数のシステムプロセスが識別される場合があります。たとえば、同じシステム上で、数値プロセス 5678 が大域ゾーンで `syslogd`、カーネルゾーンで `sendmail` を実行している場合があります。

`kzone1` で `ps` コマンドを使用してプロセス 5678 を強制終了するには、`zlogin` コマンドの後ろに `kill` コマンドを使用します。

```
global# zlogin kzone1 kill 5678
```

カーネルゾーンのゾーンパス

設計上、カーネルゾーンのゾーンパスは設定できません。これは、永続的データも、サービス可能なその他のデータも含みません。

カーネルゾーンのリソース管理機能

`max-processes` などのリソース制御は、カーネルゾーンを構成する場合に使用できません。カーネルゾーンには、大域ゾーンから独立したカーネルがあるため、カーネルゾーンの内部から実行しているプロセスは、大域ゾーンのプロセステーブルスロットを占有できません。

カーネルゾーンと不変ゾーンの操作

不変ゾーンは、読み取り専用、つまり不変のファイルシステムプロファイルを提供します。不変ゾーンは、`solaris` ブランドゾーン (非大域ゾーン内) とカーネルゾーンの両方でサポートされます。不変ゾーンの詳細は、『[Oracle Solaris ゾーン](#)の作成と使用』の「[読み取り専用ゾーンの概要](#)」を参照してください。

カーネルゾーンでのリムーバブルデバイスの管理

カーネルゾーンで仮想 CD-ROM デバイスとして機能する、リムーバブルループバックファイル `lofi` デバイスを構成できます。

▼ 仮想 CD-ROM デバイスをカーネルゾーンに追加する方法

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. 空の読み取り専用リムーバブル `lofi` デバイスを大域ゾーンに作成します。

```
# lofiadm -r
```

次の例は出力例を示しています。

```
global# lofiadm -r
/dev/lofi/1
```

3. `lofi` デバイスをカーネルゾーン構成に追加します。

```
# zonecfg -z zonename
```

次の例は、`/dev/lofi/1` にある `lofi` デバイスをカーネルゾーン `kzone1` に追加する方法を示しています。

```
global# zonecfg -z kzone1
zonecfg:kzone1> add device
zonecfg:kzone1:device> set match=/dev/lofi/1
zonecfg:kzone1:device> end
zonecfg:kzone1> exit
```

4. カーネルゾーンをリブートして、構成の変更を適用します。

```
# zoneadm -z zonename reboot
```

5. カーネルゾーンにログインします。

```
# zlogin zonename
```

6. カーネルゾーンで、デバイスのファイルシステム (`devfs`) を更新し、ハードウェア抽象化レイヤー (`hal`) を再起動して、`hal` サービスに仮想 `cdrom` デバイスが表示されるようにします。

```
zonename# devfsadm -i zvblk
zonename# svcadm restart hal
```

7. カーネルゾーン上のリムーバブルデバイスを一覧表示します。

```
zonename# rmformat -l
```

たとえば、次の例は、カーネルゾーン kzone1 のリムーバブルデバイスの一覧を示しています。

```
kzone1# rmformat -l
Looking for devices...
  1. Logical Node: /dev/rdisk/c1d0p0
     Physical Node: /zvnex/zvblk@0
     Connected Device: kz          vDisk          0
     Device Type: Removable
     Bus: <Unknown>
     Size: 16.4 GB
     Label: <Unknown>
     Access permissions: <Unknown>
  2. Logical Node: /dev/rdisk/c1d1p0
     Physical Node: /zvnex/zvblk@1
     Connected Device: kz          vCDROM      0
     Device Type: CD Reader
     Bus: <Unknown>
     Size: 0.0 MB
     Label: <Unknown>
     Access permissions: <Unknown>
```

8. 大域ゾーンで、リムーバブルループバックデバイスと関連付ける ISO イメージファイルへのパスを指定します。

```
global# lofiadm -r image-path device-path
```

次の例は、イメージパス /root/sol-11_3-repo.full.iso と lofi デバイス /dev/lofi/1 の関連付けを示しています。

```
global# lofiadm -r /root/sol-11_3-repo-full.iso /dev/lofi/1
global# lofiadm
Block Device          File                Options
/dev/lofi/1           /root/sol-11_3-repo-full.iso  Removable,ReadOnly
```

9. CD-ROM デバイスをカーネルゾーンにマウントします。

```
# mount -F hsfs device-location /mnt
```

次の例では、/dev/dsk/c1d1p0 にある仮想 CD-ROM デバイスをマウントします。

```
kzone1# mount -F hsfs /dev/dsk/c1d1p0 /mnt
```

10. 仮想 CD-ROM の使用が終わったら、カーネルゾーン内のマウントポイントからアンマウントします。

```
kzone1# umount /mnt
```

11. カーネルゾーンで、CD-ROM 仮想デバイスを取り出します。

```
kzone1# eject cdrom
```

12. ISO イメージが大域ゾーン内の lofi デバイスに関連付けられていないことを確認します。

```
# lofiadm
```

例:

```
global# lofiadm
Block Device      File      Options
/dev/lofi/1      -         Removable, Readonly
```

カーネルゾーンの補助状態の操作

カーネルゾーンでは、*auxiliary states* を使用して、大域ゾーンに補助状態の情報を通知します。デフォルトでは、カーネルゾーンに補助状態は設定されていません。補助状態が設定されるのは、デバッグおよびカーネルの保守操作を開始する場合のみです。

大域ゾーンの現在の状態およびカーネルゾーンの補助状態を表示するには、`zoneadm list -s` コマンドを使用します。

```
global# zoneadm list -s
NAME      STATUS      AUXILIARY STATE
global    running
kzone1    running
kzone2    running
kzone3    running    debugging
```

カーネルゾーンの補助状態は次のとおりです。

suspended	ゾーンは一時停止しており、次のブート時に再開されます。この状態を表示できるようにするには、ゾーンに接続する必要があります。カーネルゾーンは、移行中に <code>suspended</code> の補助状態として表示されます。 第3章「Oracle Solaris カーネルゾーンの移行」 を参照してください。
debugging	カーネルゾーンはカーネルデバッガ <code>kldb</code> 内にあります。カーネルゾーンは <code>running</code> 状態ですが、ゾーンはネットワークリクエストを処理できません。 <code>kldb</code> と対話するには、ゾーンコンソールに接続する必要があります。ゾーンコンソールに接続する方法については、『 Oracle Solaris ゾーンの作成と使用 』の第4章、「 非大域ゾーンへのログインについて 」を参照してください。
panicked	ゾーンは <code>running</code> 状態ですが、パニックが発生しています。ホストシステムは影響を受けません。パニックが発生している補助状態のカーネルゾーンにログインするには、ゾーンコンソールアクセスが必要です。
migrating-out	ゾーンは完全に実行中ですが、別のシステムに移行中です。

migrating-in ゾーンはシステムでブートされ、移行イメージを受信していません。移行が完了するまではまだ完全に実行されていません。

ゾーンの状態については、『[Oracle Solaris ゾーン](#)の紹介』の第1章、「[Oracle Solaris ゾーンの概要](#)」を参照してください。カーネルゾーンの補助状態の詳細は、[solaris-kz\(5\)](#)のマニュアルページを参照してください。カーネルデバッグの詳細は、[kmdb\(1\)](#)のマニュアルページを参照してください。

入れ子にされたゾーンの管理

入れ子にされたゾーンとは、カーネルゾーン内からインストールおよびブートされる非大域ゾーンです。入れ子にされたゾーンでは、カーネルゾーンは大域ゾーンとして機能します。入れ子にされたゾーンは、新しい **solaris** ブランドゾーン、または移行された **solaris** または **solaris10** ブランドゾーンになります。カーネルゾーンは入れ子にされたゾーンとしてサポートされません。

入れ子にされたゾーンの要件は次のとおりです。

- オペレーティングシステム** 入れ子にされたゾーンはすべて、Oracle Solaris 11.2 以降をサポートしている必要があります。
- Oracle Solaris 11 または Oracle Solaris 11.1 を実行している **solaris** ブランドゾーンは、Oracle Solaris 11.2 に更新する必要があります。システムソフトウェアパッケージの更新については、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の第3章、「[ソフトウェアパッケージのインストールおよび更新](#)」を参照してください。
 - **solaris10** ブランドゾーンを、Oracle Solaris 11.2 以上を実行する非大域ゾーンに移行できます。**solaris10** ゾーンを Oracle Solaris 11.2 システムに移行する方法の手順については、『[Oracle Solaris 10 ゾーンの作成と使用](#)』の「[別のホストへの solaris10 ブランドゾーンの移行](#)」を参照してください。
- ネットワーク構成** ネストされたゾーンとして実行する **solaris** または **solaris10** ブランドゾーンでは、排他的 IP または共有 IP を使用できます。排他的 IP 構成が必要な場合、追加の MAC アドレスを許可するようにカーネルゾーンを構成する必要があります。
- システムリソース** 入れ子にされたゾーンは、カーネルゾーンで利用可能なシステムリソースのみを使用できます。これらのリソースには、仮想ディスクおよび iSCSI ディスクが含まれます。
- クローニング** 入れ子にされた構成を含むカーネルゾーンがクローニングされた場合、外部のカーネルゾーンのみがクローニングされます。カー

ネルゾーンの内部のゾーンは、ゾーンのクローニングプロセス中にクローニングされません。50 ページの「カーネルゾーンのクローニング」を参照してください。

▼ 複数の MAC アドレスをカーネルゾーンに追加する方法

この手順では、2つの自動的に生成された MAC アドレスをカーネルゾーンに追加する方法を示します。

1. 管理者になります。

詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。

2. 新しい MAC アドレスを追加します。

```
global # zonecfg -z kz0
zonecfg:kz0> add anet
zonecfg:kz0:anet> add mac
zonecfg:kz0:anet:mac> end
zonecfg:kz0:anet> add mac
zonecfg:kz0:anet:mac> end
zonecfg:kz0:anet> end
zonecfg:kz0> exit
global #
```

3. カーネルゾーンをブートするか、実行中のカーネルゾーンに変更を適用します。

```
global # zoneadm -z kz0 apply
zone 'kz0': Checking: Adding anet id=1
zone 'kz0': Applying the changes
```

4. (オプション) カーネルゾーンにログインし、新しい MAC アドレスを表示します。

```
global # zlogin kz0
kz0# dladm show-phys -m net1
LINK          SLOT      ADDRESS          INUSE CLIENT
net1          primary  2:8:20:42:cf:83  yes  net1
              1        2:8:20:f4:e1:b1  no   --
              2        2:8:20:38:67:f3  no   --
```

入れ子にされたゾーンおよび新しい非大域ゾーンの構成

zonecfg および zoneadm コマンドを使用して、カーネルゾーン内から新しい solaris ブランドゾーンを構成、インストール、およびブートできます。例:

```
kzone1# zonecfg -z zone1
```

```
Use 'create' to begin configuring a new zone.
zonecfg:zone1> create -t SYSsolaris
zonecfg:zone1> commit
zonecfg:zone1> exit
```

非大域ゾーンの計画、構成、およびインストールの詳細は、『[Oracle Solaris ゾーンの実装と使用](#)』を参照してください。

カーネルゾーンのホストデータおよびホスト ID

カーネルゾーンのブート可能な各デバイスには、ホストデータとして知られる状態情報が含まれています。カーネルゾーンのホストデータは、次のようなカーネルゾーンの状態情報をモニターします。

- ゾーンの使用
- [34 ページの「suspend リソースの構成」](#)で説明されている、ゾーンの一時停止
- カーネルゾーンクロックと大域ゾーンクロックの間の時間のオフセット
- OpenBoot 変数 (SPARC のみ)

カーネルゾーンのホストデータは、カーネルゾーン suspend イメージに使用されるものと同じ暗号化鍵を使用する、高度な暗号化標準の AES-128-CCM で暗号化および認証されます。

カーネルゾーンが構成またはブートされた場合、ホストデータが読み取られて、カーネルゾーンのブートストレージがほかのシステムで使用中心かどうかを判別されます。ブートストレージが別のシステムで使用中心の場合、カーネルゾーンは `unavailable` 状態になり、ブートストレージを使用しているシステムを示すエラーメッセージが表示されます。例:

```
global# zoneadm -z kzone1 attach
zone 'kzone1': error: ERROR: zone kzone1 is in use by host with hostid 848611d4
zone 'kzone1': error:      last known state: installed
zone 'kzone1': error:      hostname: global2
zone 'kzone1': error: boot environment name: solaris-1
zone 'kzone1': error: boot environment uuid: 69ed2e6a-e25a-6d36-e022-ed7261ed8899
zone 'kzone1': error:      last update time: Sun Apr 13 20:08:13 2014
zone 'kzone1': error: To fix, detach the zone from the other host then attach it to this
host
zone 'kzone1': error: If the zone is not active on another host, attach it with
zone 'kzone1': error:      zoneadm -z kzone1 attach -x force-takeover
```

ブートストレージが別のシステムで使用されていない場合、`zoneadm attach -x force-takeover` コマンドを使用してカーネルゾーンを修復できます。



注意 - ホストデータの引き継ぎまたは再初期化を強制すると、ゾーンがほかのシステムで使用中心かどうかの検出が不可能になります。同じストレージを参照するゾーンの複数インスタンスを実行すると、ゾーンのファイルシステムの修復不能な破壊につながります。

ゾーンの暗号化鍵にアクセスできない場合、ホストデータと一時停止イメージすべてを読み取ることができません。そのような状況では、ゾーンを準備またはブートするすべての試行によって、ゾーンが `unavailable` 状態になります。ゾーンの暗号化鍵を復元できない場合、`zoneadm attach -x initialize-hostdata` コマンドを使用して新しい暗号化鍵およびホストデータを生成します。

カーネルゾーンの移行中の暗号化鍵の損失を防ぐには、ソースシステム上で `zonecfg export` コマンドを使用して、ターゲットシステム上で使用するコマンドファイルを生成します。例:

```
global# zonecfg -z kzone1 export -f /net/.../kzone1.cfg
global# zonecfg -z kzone1 -f /net/.../kzone1.cfg
```

カーネルゾーンのブートローダーの操作

カーネルゾーンのブートローダーは、カーネルゾーンのブート操作を管理します。ブートローダーを起動するには、カーネルゾーンが準備またはインストールされた状態である必要があります。カーネルゾーンのブートローダーを使用すると、次の操作を実行できます。

- 使用可能なブート環境の一覧表示
- 代替ブート環境へのゾーンのブート

カーネルゾーンのブートローダーを起動するには、`zoneadm boot` コマンドを使用します。カーネルゾーンのブートローダーを起動する場合は、ゾーンコンソールも起動する必要があります。ブートローダーの出力はゾーンコンソールに表示されます。

注記 - ゾーンコンソールから終了するコマンドシーケンスは `~` です。詳細は、『[Oracle Solaris ゾーンの実装と使用](#)』の「[ゾーンコンソールへのログイン方法](#)」を参照してください。

オペレーティングシステムレベルでのブート環境の作成および管理については、『[Oracle Solaris 11.3 ブート環境の作成と管理](#)』の第1章、「[ブート環境の管理の概要](#)」を参照してください。ゾーンおよびブート環境の管理の詳細については、『[Oracle Solaris 11.3 ブート環境の作成と管理](#)』の第2章、「[beadm でのゾーンのサポート](#)」を参照してください。

▼ カーネルゾーンで代替ブート環境を指定する方法

1. 管理者になります。
詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。

2. ゾーンコンソールにログインします。

```
# zlogin -C zonename
```

たとえば、kzone1 でコンソールにログインするには、次のようにします。

```
global# zlogin -C kzone1
```

3. 別の端末ウィンドウで、使用可能なカーネルゾーンのブート環境を一覧表示します。

```
# zoneadm -z zonename boot -- -L
```

次の例は出力例を示しています。

```
global# zoneadm -z kzone2 boot -- -L
[Connected to zone 'kzone2' console]
1 kz-130118 (rpool/ROOT/kz-130118)
2 kz-1 (rpool/ROOT/kz-1)
3 solaris-5 (rpool/ROOT/solaris-5)
4 solaris-7 (rpool/ROOT/solaris-7)
Select environment to boot: [ 1 - 4 ]:
```

4. 選択したブート環境にブートします。

```
# zoneadm -z zonename boot -- -Z boot-environment
```

例:

```
global# zoneadm -z kzone1 boot -- -Z rpool/ROOT/solaris-backup-1
```

例 35 SPARC ベースシステムでの代替ブート環境の選択とブート

次の例は、カーネルゾーン kzone1 の代替ブート環境のゾーンコンソール出力を示しています。カーネルゾーンのホストハードウェアは SPARC ベースシステムです。

```
[Connected to zone 'kzone1' console]
NOTICE: Entering OpenBoot.
NOTICE: Fetching Guest MD from HV.
NOTICE: Starting additional cpus.
NOTICE: Initializing LDC services.
NOTICE: Probing PCI devices.
NOTICE: Finished PCI probing.
```

```
SPARC T4-2, No Keyboard
Copyright (c) 1998, 2014, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.36.0.build_05, 2.0000 GB memory available, Serial #1845652596.
Ethernet address 0:0:0:0:0:0, Host ID: 6e026c74.
```

```
Boot device: disk0 File and args: -L
1 Oracle Solaris 11.2 SPARC
2 bootenv123
3 bootenv456
Select environment to boot: [ 1 - 3 ]: 2
```

```
To boot the selected entry, invoke:
boot [<root-device>] -Z rpool/ROOT/bootenv123
```

```

Program terminated
ok boot -Z rpool/ROOT/bootenv123

[NOTICE: Zone rebooting]
NOTICE: Entering OpenBoot.
NOTICE: Fetching Guest MD from HV.
NOTICE: Starting additional cpus.
NOTICE: Initializing LDC services.
NOTICE: Probing PCI devices.
NOTICE: Finished PCI probing.

SPARC T4-2, No Keyboard
Copyright (c) 1998, 2014, Oracle and/or its affiliates. All rights reserved.
OpenBoot 4.36.0.build_05, 2.0000 GB memory available, Serial #1845652596.
Ethernet address 0:0:0:0:0:0, Host ID: 6e026c74.

...
Hostname: kzone1
kzone1 console login:

```

例 36 x86 ベースシステムでの代替ブート環境の選択とブート

次の例は、カーネルゾーン `kzone1` の代替ブート環境のゾーンコンソール出力を示しています。カーネルゾーンのホストハードウェアは x86 システムです。

```

[Connected to zone 'kzone1' console]
1 boot-2 (rpool/ROOT/boot-2)
2 Oracle Solaris 11.2 x86 (rpool/ROOT/solaris)
3 boot-1 (rpool/ROOT/boot-1)
Select environment to boot: [ 1 - 3 ]:2
Boot device: disk0 File and args:
reading module /platform/i86pc/amd64/boot_archive...done.
reading kernel file /platform/i86pc/kernel/amd64/unix...done.
SunOS global 5.11 11.2 i86pc i386 i86pc
Copyright (c) 1983, 2014, Oracle and/or its affiliates. All rights reserved.
Hostname: kzone1
...
kzone1 console login:

```

カーネルゾーンのライブゾーン再構成

ライブゾーン再構成を使用すると、ゾーンの実行中に `solaris-kz` ゾーンのライブ構成を再構成またはそのレポートを作成できます。この機能の詳細については、『[Oracle Solaris ゾーンの作成と使用](#)』の第 6 章、「[ライブゾーン再構成](#)」を参照してください。

NFS ストレージ URI とカーネルゾーン

Oracle Solaris カーネルゾーンの NFS ストレージ URI (Uniform Resource Identifier) を構成できます。ストレージ URI は、さまざまなノード全体で共有ストレージオブジェクト

トを一意に識別するために使用されます。共有ストレージを使用すると、ゾーン内の共有ストレージリソースに透過的にアクセスして管理できます。

NFS ストレージ URI はカーネルゾーンでのみサポートされています。

NFS ストレージ URI の構文と使用法

NFS URI は、指定された NFS ファイル上に作成された `lofi` デバイスに基づいてオブジェクトを指定します。NFS ファイルは、ユーザーとグループから取得された資格情報を使用してアクセスされます。ユーザーおよびグループは、ユーザー名またはユーザー ID で指定できます。ホストは IPv4 アドレス、IPv6 アドレス、またはホスト名で指定できます。IPv6 アドレスは、角括弧で囲む必要があります。

`nfs-share-path` の値は、通常のバッキングストアファイルが含まれているホストサーバーからの `nfs` エクスポートディレクトリである必要があります。NFS ストレージ URI の構文は次のとおりです。

```
nfs:///user:group@host[:port]/nfs-share-path/file
```

次の例に、URI 構文の使用方法を示します。

- `nfs://admin:staff@host/export/test/nfs_file`
- `nfs://admin:staff@host:1000/export/test/nfs_file`

NFS ストレージ URI は `suriadm` コマンドによって管理できます。トラブルシューティングおよび回復には、`suriadm` プロパティ `mountpoint-prefix=/system/volatile/zones/zonename` を使用します。詳細は、[suriadm\(1M\)](#) のマニュアルページまたは『[Oracle Solaris ゾーンの作成と使用](#)』の「[ストレージ URI と共有ストレージリソースの管理](#)」を参照してください。

カーネルゾーン内のコアファイル

カーネルゾーンのプロセスが突然終了した場合は、生成されるコアファイルが、カーネルゾーン上の `dumpadm` コマンドによって定義されている場所に保存されます。

カーネルゾーン内にコアダンプの生成を避ける状況で、カーネルゾーンがクラッシュすることがあります。そのような場合にカーネルゾーンのコアダンプを生成してアクセス可能にするには、大域ゾーンで `coreadm` コマンドを使用して、これらのコアダンプの場所を有効にし、指定します。

詳細は、`dumpadm` および `coreadm` のマニュアルページを参照してください。

索引

あ

- 一般的なゾーンの概念, 12
- 入れ子にされたゾーン, 70
- インストールイメージ
 - カーネルゾーンのインストールに使用, 48
- ウォーム移行, 54

か

- カーネルゾーン
 - Silicon Secured Memory の有効化, 38
 - カーネルゾーン構成, 17
 - カーネルゾーンのアンインストール, 49
 - カーネルゾーンの移行, 53, 53, 54, 54, 55, 55
 - カーネルゾーンの一時停止, 34
 - カーネルゾーンのインストール
 - Automated Installation (AI) マニフェスト, 44
 - sysconfig プロファイル, 44
 - インストールイメージ, 48
 - 直接インストール, 42
 - カーネルゾーンのクローニング, 50
 - カーネルゾーンの構成, 17
 - カーネルゾーンの再開, 34
 - カーネルゾーンの停止処理, 49
 - カーネルゾーンのデフォルトのインストール方法, 42
 - カーネルゾーンのブートローダー, 73
 - カーネルゾーンのルートディスク, 22
 - カーネルゾーンリソースの構成, 18
- 階層ゾーン
 - 構成, 70
 - 要件, 70
- 仮想 LAN, 31
 - 動的な MAC アドレスと VLAN ID, 33
- コールド移行, 53

さ

- シャドウ VNIC, 30
- 重複プロセス ID
 - カーネルゾーン, 66
- ストレージデバイス, 25
- ストレージデバイスの追加, 25
- セキュアなライブ移行, 62
- ゾーン
 - ブランド 参照 solaris-kz ブランドゾーン
- ゾーンパス, 66
- ソフトウェア要件, 13

た

- 直接インストール, 42

な

- ネットワークデバイス
 - 削除, 26
 - 追加, 26
- ネットワークデバイスの追加, 26
- ネットワークデバイス ID, 26

は

- ハードウェア要件, 13
- ブート環境、指定, 73
- ブートローダー, 73
- 不変カーネルゾーンの構成, 66
- 不変ゾーン, 66
- ブランド, 11
- ブランドゾーン, 11
- プロセス ID

- カーネルゾーンでの可視性, 65
- 補助状態, 69
- ホストデータ, 72
- ホストでのサポートの確認, 15
- ホスト要件, 13
- ホスト ID, 72

ま

- メモリー
 - カーネルゾーン用に予約, 16
 - 管理, 21
 - ページサイズ, 21
- メモリーの追加, 21

や

- 読み取り専用カーネルゾーン, 66
- 読み取り専用カーネルゾーンの構成, 66

ら

- ライブ移行, 55
 - Oracle Solaris 11.3 ターゲットシステムへの, 24
 - root 以外のユーザーによる, 57
- リムーバブルデバイスの構成, 67
- ルートディスクサイズの増加, 25

A

- add device リソースプロパティ, 25
- ADI, 38
 - 参照 SSM
- anet リソース, 26
- archiveadm コマンド, 50
- Automated Installation (AI) マニフェスト
 - カーネルゾーンのインストールに使用, 44

B

- bootpri リソースプロパティ, 25
- BrandZ, 11

C

- capped-memory リソースプロパティ, 21
- CPU
 - 管理, 19

D

- dedicated-cpu リソースプロパティ, 19

H

- host-compatible プロパティ, 38

K

- kz-migr サービス, 56

L

- lofi デバイス, 67

N

- net リソース, 26

O

- Oracle Solaris カーネルゾーン, 12
 - 参照 カーネルゾーン、solaris-kz ブランドゾーン、オペレーティングシステム要件
 - 関連概念, 12
 - 使用の利点, 11

P

- pagesize-policy プロパティ, 21, 23

S

- set_user_reserve.sh スクリプト, 16

solaris-kz ブランドゾーン, 11
SPARC ファームウェア要件, 13
SR-IOV, 28
SSM, 38
sysconfig プロファイル
 カーネルゾーンのインストール, 44

U

user_reserve_hint_pct チューニング可能パラ
メータ, 16

V

virtinfo コマンド, 15
virtual-cpu リソースプロパティ, 20
VLAN 対応のカーネルゾーン, 31
 動的な MAC アドレスと VLAN ID, 33

X

x86 BIOS 要件, 13

Z

ZFS ARC キャッシュ要件, 13
ZFS ARC チューニング, 16
Zone Migration 権利プロファイル, 57
zoneadm boot コマンド, 73
zoneadm clone コマンド, 50
zoneadm halt コマンド, 49
zoneadm install コマンド, 41
zoneadm list -s コマンド, 69
zoneadm migrate コマンド, 57, 57
zoneadm reboot コマンド, 49
zoneadm shutdown コマンド, 49
zoneadm suspend コマンド, 34
zoneadm uninstall コマンド, 49
zonecfg bwshare anet プロパティ, 31

