

Oracle® Solaris カーネルゾーンの作成と使用

ORACLE®

Part No: E75325-01
2018 年 8 月

Part No: E75325-01

Copyright © 2014, 2018, 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法	9
1 Oracle Solaris カーネルゾーンの計画	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
カーネルゾーンのセキュリティー対策	13
Oracle Solaris カーネルゾーンのソフトウェアおよびハードウェア要件	14
Oracle Solaris カーネルゾーンのソフトウェア要件	14
Oracle Solaris カーネルゾーンのハードウェア要件	14
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	16
カーネルゾーン用にメモリーを予約するためのホスト ZFS ARC の チューニング	17
SPARC: カーネルゾーンでのソフトウェアインシリコン機能	18
2 Oracle Solaris カーネルゾーンの構成	21
Oracle Solaris カーネルゾーンの構成	21
▼ カーネルゾーンを構成する方法	21
カーネルゾーンのリソースの管理について	22
カーネルゾーンの CPU の管理	23
カーネルゾーンのメモリーの管理	25
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	28
カーネルゾーンのネットワークデバイスと構成の管理	29
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	31
suspend リソースタイプの構成	36
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	37
IPoIB とカーネルゾーンの操作	38
カーネルゾーンでの仮想 LAN の構成	38
カーネルゾーンでの動的な MAC アドレスと VLAN ID の使用	40

3 Oracle Solaris カーネルゾーンのインストール、シャットダウン、およびクローニング	
ローニング	43
カーネルゾーンのインストール	43
カーネルゾーンの直接インストール	44
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	46
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	49
インストールイメージを使用したカーネルゾーンのインストール	51
カーネルゾーンのアンインストール	53
カーネルゾーンのシャットダウン、停止、およびリブート	53
カーネルゾーンのクローニング	54
4 カーネルゾーンのライブゾーン再構成	57
ライブゾーン再構成の実行	57
▼ 実行中のゾーンのライブ構成を検査する方法	57
▼ ライブゾーン構成の影響を確認する方法	58
▼ ライブゾーンに永続的な構成変更を加える方法	58
▼ 実行中のゾーンに一時的な変更を加える方法	60
▼ ライブゾーン構成の変更の確定中に障害から回復する方法	61
5 Oracle Solaris カーネルゾーンの移行	63
カーネルゾーンの移行について	63
カーネルゾーン移行の実行に必要な権利	64
使用する移行方式の決定	66
カーネルゾーンの移行の要件	67
カーネルゾーンのすべての移行方式での要件	67
カーネルゾーンのウォーム移行およびライブ移行での追加要件	68
カーネルゾーンのライブ移行での追加要件	69
カーネルゾーンの移行の準備	69
zoneadm migrate コマンドについて	70
移行および互換性のある構成について	71
カーネルゾーンを異なる CPU または OS バージョンを持つシステムに 移行するための準備	72
セキュアな移行	75
コールド移行を使用したカーネルゾーンの移行	76
▼ コールド移行を使用してカーネルゾーンを移行する方法	77
ウォーム移行またはライブ移行用のサービスの有効化	78
▼ カーネルゾーンの移行に必要なサービスを確認および有効化する方 法	79

ウォーム移行を使用したカーネルゾーンの移行	80
▼ 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	81
ウォーム移行後のカーネルゾーンの再開について	83
ライブ移行を使用したカーネルゾーンの移行	83
▼ ライブ移行を使用してカーネルゾーンを移行する方法	84
6 ターゲットホストへの Oracle Solaris カーネルゾーンの退避	87
カーネルゾーン退避について	87
カーネルゾーン退避のステップ	87
sysadm ユーティリティと Oracle Solaris カーネルゾーン	88
カーネルゾーン退避の要件	89
カーネルゾーン退避用のターゲットホストの設定	90
カーネルゾーン退避の準備としての保守モードの設定	91
カーネルゾーンの退避	92
▼ 実行中のカーネルゾーンを退避する方法	92
ゾーン退避のステータスの確認	94
カーネルゾーン退避後の保守モードの終了	94
退避されたカーネルゾーンを元のシステムに戻す	94
カーネルゾーン退避の例	95
7 Oracle Solaris カーネルゾーンの管理	97
カーネルゾーン環境での作業	97
カーネルゾーンのプロセス情報の表示	98
カーネルゾーンと大域ゾーンで重複するプロセス ID	98
カーネルゾーンのゾーンパス	98
カーネルゾーンでのリソース管理	98
不変カーネルゾーンの操作	99
カーネルゾーンでのリムーバブルデバイスの管理	99
▼ 仮想 CD-ROM デバイスをカーネルゾーンに追加する方法	99
大域ゾーンから確認できるカーネルゾーンの補助状態	101
カーネルゾーン内の非大域ゾーンの管理	102
カーネルゾーン内のネイティブゾーンの要件	102
▼ 複数の MAC アドレスをカーネルゾーンに追加する方法	103
カーネルゾーン内での新しい solaris ゾーンの構成	104
カーネルゾーンのホストデータおよびホスト ID	104
ホストデータからのストレージ情報	104
暗号化鍵とホストデータ	105
カーネルゾーンのブートローダーの起動	106

▼ カーネルゾーンで代替ブート環境を指定する方法	106
NFS ストレージ URI とカーネルゾーン	108
カーネルゾーン内のコアファイル	109
索引	111

このドキュメントの使用方法

- **概要** – Oracle Solaris カーネルゾーンを計画、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。Oracle Solaris ゾーンの知識。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=E75431-01> 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。

Oracle Solaris カーネルゾーンの計画

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) のハードウェア、ソフトウェア、およびセキュリティー要件を確認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とソフトウェアサポートを確認する手順、およびカーネルゾーンのセキュアな管理を構成する手順を説明します。

この章で扱う内容は、次のとおりです。

- 11 ページの「Oracle Solaris カーネルゾーンについて」
- 12 ページの「カーネルゾーンと一般的なゾーンの概念」
- 13 ページの「カーネルゾーンのセキュリティー対策」
- 14 ページの「Oracle Solaris カーネルゾーンのソフトウェアおよびハードウェア要件」
- 18 ページの「カーネルゾーンでのソフトウェアインシリコン機能」

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 ゾーン) とは、大域ゾーンとは別の専用のカーネルとオペレーティングシステムを備えた非大域ゾーンです。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、セキュリティーが強化されます。

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、カーネルゾーンは、大域ゾーン (カーネルゾーンホスト) とソフトウェア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、カーネルゾーン内部から完全に機能します。詳細は、『Oracle Solaris 11.4 でのソフトウェアの追加とシステムの更新』の「[非大域ゾーンで動作するオプション](#)」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、installed 状態のカーネルゾーンではサポートされていません。カーネルゾーンが configured または unavailable 状態のときに zonecfg コマンドを使用することによってのみ、カーネルゾーンの名前を変更できます。

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。

カーネルゾーンは、ブランドゾーンフレームワークの一部です。詳細は、[brands\(7\)](#) 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、『[Introduction to Oracle Solaris Zones](#)』の第1章、『[Oracle Solaris Zones Introduction](#)』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

このガイドでは、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 ゾーン紹介](#)』および『[Oracle Solaris ゾーン作成と使用](#)』を参照してください。

カーネルゾーンのセキュリティ対策

カーネルゾーン、非大域ゾーン、および大域ゾーンは、同じようなセキュリティ機能を備えています。IPsec と IKE によってネットワークを保護し、権利と監査によってリソースの不正使用を防ぎ、不変ゾーンによって管理セキュリティを強化しています。ゾーンにおけるこれらのセキュリティ対策の詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Security Measures for a System With Non-Global Zones](#)」を参照してください。

カーネルゾーンには、バリファイドブートと呼ばれる追加のセキュリティ対策もあります。詳細は、[37 ページの「バ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保護」](#) および『[Oracle Solaris 11.4 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保護](#)』の「[バリファイドブートの使用](#)」を参照してください。

ゾーンを管理する権利を root 以外のユーザーに割り当てることは、一般的なセキュリティ対策です。デフォルトでは、大域ゾーンの管理者 (root) はすべてのカーネルゾーンを管理できますが、root はそれらの権利を配布することもできます。

カーネルゾーンに適用されるゾーンの権利プロファイルとゾーン内の admin リソースの概要については、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」を参照してください。

注記 - このガイドに記載されている例と手順は、ゾーンが root 以外のユーザーによって管理されていることを前提とします。通常、root 以外のユーザーが権利を使ってコマンドを実行するには、ゾーンの管理コマンドの前に pfbash または pfexec コマンドを付けます。詳細は、[pfexec\(1\)](#) のマニュアルページを参照してください。

権利に関する背景と詳細については、次を参照してください。

- 『[Oracle Solaris 11.4 でのユーザーとプロセスのセキュリティ保護](#)』の第 1 章、「[権利を使用したユーザーとプロセスの制御について](#)」
- 『[Oracle Solaris 11.4 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](#)」
- 『[Oracle Solaris Zones Configuration Resources](#)』の「[admin Resource Type](#)」

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

Oracle Solaris には、カーネルゾーンに関して Oracle Solaris リリースバージョンの要件とハードウェア要件があります。また、ゾーンの移行にはハードウェアとソフトウェアの互換性が必要です。

Oracle Solaris カーネルゾーンのソフトウェア要件

Oracle Solaris 11.4 でサポートされているすべてのカーネルゾーン機能を使用するには、カーネルゾーンとカーネルゾーンホストの両方で Oracle Solaris 11.4 オペレーティングシステムが動作している必要があります。

カーネルゾーンの機能は、カーネルゾーンホストの機能に制限されます。たとえば、Oracle Solaris 11.3 システムでホストされた Oracle Solaris 11.4 が動作しているカーネルゾーンでは、Oracle Solaris 11.3 バージョンによってサポートされている機能に制限されます。この場合、カーネルゾーンのライブ移行はサポートされません。

Oracle Solaris カーネルゾーンのハードウェア要件

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 M8 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。
- 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。
- Fujitsu M10/SPARC M10 サーバー。Fujitsu M10 システムのプロダクトノート of ファームウェア要件 (使用している構成に適したもの) に従ってください。
- Oracle Solaris 11.3 以上がインストールされている Fujitsu SPARC M12 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。

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 \(http://www.oracle.com/technetwork/](http://www.oracle.com/technetwork/)

[systems/patches/firmware/release-history-jsp-138416.html](#))に関する Web ページを参照してください。

x86 システム:

- Nehalem またはそれ以降のプロセッサを搭載した Intel ベースシステム
- Barcelona またはそれ以降のプロセッサを搭載した AMD ベースシステム
- 互換性のある CPU 上のマイクロコードレベル
- BIOS では、次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 システムと x86 システムにはどちらも次が必要です。

- 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 11.4](#)でのソフトウェアの追加とシステムの更新』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。
- 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、コマンド `svcadm restart rad` で RAD SMF サービスを手動で再起動する必要があることに注意してください。
- 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。17 ページの「[カーネルゾーン用にメモリーを予約するためのホスト 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、次を使用してインストールできます。

- 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
- 統合アーカイブ

`-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。テキストインストーラと自動インストーラを使用すると、サポートされている任意の Oracle Solaris バージョンをイン

ストールできません。Oracle Solaris 11.2 はカーネルゾーンでサポートされている最初の Oracle Solaris バージョンです。

Oracle Solaris カーネルゾーンは、Oracle VM Server for SPARC (旧称: Sun 論理ドメイン) 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、SPARC M6、SPARC M7、または SPARC M8 システムでは 512、Fujitsu M10 と Fujitsu SPARC M12 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 上では実行できません。

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 11.4 ご使用にあたって](#)』の「[カーネルゾーンに関する問題](#)」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。[68 ページの「カーネルゾーンのウォーム移行およびライブ移行での追加要件」](#)を参照してください。

ヒント - 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、非大域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でください。

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、カーネルゾーンホストが [14 ページの「Oracle Solaris カーネルゾーンのソフトウェアおよびハードウェア要件」](#) を満たしていることを確認する必要があります。

▼ システムでカーネルゾーンをサポートできることを確認する方法

1. **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 11.4 sun4v sparcsun4v
```

2. カーネルゾーンのブランドパッケージ `brand/brand-solaris-kz` がインストールされていることを確認します。

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

この例は、カーネルゾーンのブランドパッケージがシステム `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  11.4-11.4.0.0.1.10.1  i--
```

3. カーネルゾーンが大域ゾーンでサポートされていることを確認します。

```
$ virtinfo
```

この出力例は、カーネルゾーンがシステム `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
logical-domain parent
non-global-zone supported
kernel-zone   supported
```

参照 詳細は、[virtinfo\(8\)](#) のマニュアルページを参照してください。

カーネルゾーン用にメモリーを予約するためのホスト 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には、カーネルゾーンホストで `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。このパラメータは、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。そのカーネルゾーンホストから見れば、カーネルゾーンはアプリケーションです。ARC の増加を制限することで、アプリケーション (カーネルゾーンとカーネルゾーン内で実行されるアプリケーションを含む) に使用できるメモリーをより多く残すことができます。



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、メモリー不足障害が発生する可能性があります。

システム上の ZFS ARC を制限するには、カーネルゾーンホストで `user_reserve_hint_pct` パラメータを設定します。 `set_user_reserve.sh` と呼ば

れるスクリプトを使用して、このパラメータの値を 80 に設定することをお勧めします。このスクリプトは、パラメータを動的に調整します。

カーネルゾーンホストで実行され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。

たとえば、global というカーネルゾーンホストでスクリプトを実行する場合:

```
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 : user
set user_reserve_hint_pct=80
```

注記 - スクリプトを実行すると、実行中のシステムで user_reserve_hint_pct パラメータがチューニングされます。このパラメータを、リブート後も持続するようにするには、パラメータを /etc/system 内に設定します。

SPARC: カーネルゾーンでのソフトウェアインシリコン機能

Oracle ソフトウェアインシリコンの機能は、Oracle Solaris 11.3 や Oracle Solaris 11.4 が動作している、Oracle の SPARC M7 プロセッサベースのサーバーで使用できます。ソフトウェアインシリコンテクノロジーには、Silicon Secured Memory (SSM) と Data Analytics Accelerator (DAX) があります。SSM を使用すると、Application Data Integrity (ADI) が有効になります。

Oracle ソフトウェアインシリコン機能の詳細は、[ソフトウェアインシリコンを使用した包括的な Oracle ソリューションのセキュリティとパフォーマンス上の利点に関するホワイトペーパー](#)を参照してください。

SSM と DAX は、それらがホストシステムの CPU やカーネルゾーンのオペレーティングシステムでサポートされていても、カーネルゾーンではデフォルトで無効になります。このデフォルトは、こうした機能をサポートしていない以前のシステムとの間のカーネルゾーンのウォーム移行やライブ移行に役立つように選ばれたものです。

Oracle Solaris 11.3 または Oracle Solaris 11.4 が動作しているカーネルゾーンでこれらの機能を有効にするには、`host-compatible` プロパティを設定します。

- SSM のみを有効にするには、`host-compatible=adi` を設定します。
- DAX、仮想アドレスマスキング (VA マスキング) および SSM を有効にするには、`host-compatible=level1` を設定します。

注記 - 移行クラスとホストの互換性レベルの両方で有効にされた機能のみがカーネルゾーンに表示されます。SSM または DAX を使用する場合は、`cpu-arch` プロパティを移行クラスに設定しないでください。

SSM や DAX が使用可能でない以前の SPARC ベースシステムまたは以前のバージョンの Oracle Solaris ソフトウェアにカーネルゾーンを移行するには、移行を開始する前に、まず次の構成変更を行う必要があります。

1. `host-compatible` プロパティを互換性のある値に設定するか、そのプロパティをクリアして、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します。
2. 以前の SPARC ベースシステムに移行するように `cpu-arch` プロパティを設定します。詳細は、72 ページの「[カーネルゾーンを異なる CPU または OS バージョンを持つシステムに移行するための準備](#)」を参照してください。

詳細は、『[Oracle Solaris Zones Configuration Resources](#)』の「[Kernel Zone Migration Class and Host Compatibility Level \(solaris-kz Only\)](#)」を参照してください。

例 1 カーネルゾーンでの SSM の有効化

この例では、`host-compatible` が設定されているかどうかを確認してから、そのプロパティを `adi` に設定し、ゾーンをブートします。`info` サブコマンドでは、明示的に設定されていないプロパティに関する情報は、たとえそのプロパティにデフォルト値がある場合でも一切表示されません。

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

例 2 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

この例は、SPARC T5 システムのカーネルゾーンで SSM を有効にする試みを示しています。SPARC T5 では、SSM がサポートされていません。カーネルゾーンのブートを試みると、エラーが検出されます。

```
global$ pfbash 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 3 カーネルゾーンでの DAX の有効化

この例は、何も出力されないことによって、カーネルゾーン kzone1 で host-compatible プロパティが設定されておらず、DAX、VA マスキング、および SSM を有効にするためにそのプロパティを level1 に設定して、ゾーンをブートすることを示しています。

```
global$ pfbash zonecfg -z kzone1
zonecfg:kzone1> info host-compatible
zonecfg:kzone1> set host-compatible=level1
zonecfg:kzone1> exit
global$ zonecfg -z kzone1 boot
```

例 4 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、カーネルゾーン kzone1 で host-compatible プロパティをクリアしてから、ゾーンをリブートします。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する場合は、移行の前に、[72 ページの「カーネルゾーンを異なる CPU または OS バージョンを持つシステムに移行するための準備」](#)の説明に従って、cpu-arch プロパティをリセットする必要もあります。

```
global$ pfbash zonecfg -z kzone1 clear host-compatible
global$ zoneadm -z kzone1 reboot
```

Oracle Solaris カーネルゾーンの構成

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) を構成する方法について説明します。この章の内容は次のとおりです。

- 21 ページの「Oracle Solaris カーネルゾーンの構成」
- 22 ページの「カーネルゾーンのリソースの管理について」

Oracle Solaris カーネルゾーンの構成

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 Zones Configuration Resources](#)』の「[zonecfg Templates](#)」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、『[Creating and Using Oracle Solaris Zones](#)』の第 1 章、「[Planning a Non-Global Zone Configuration](#)」を参照してください。

▼ カーネルゾーンを構成する方法

デフォルトのカーネルゾーンテンプレート SYSsolaris-kz を使用するカーネルゾーンを構成するには、この手順を実行します。

- 始める前に
- 重要なカーネルゾーンリソースに関する情報など、カーネルゾーン構成の要件やガイドラインを確認します。第 1 章「[Oracle Solaris カーネルゾーンの計画](#)」を参照してください。
 - ホストシステム上で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、およびメモリ構成を確認します。16 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。
 - カーネルゾーンをホストするシステム上で user_reserve_hint_pct チューニング可能パラメータを設定します。17 ページの「[カーネルゾーン用にメモリーを予約するためのホスト ZFS ARC のチューニング](#)」を参照してください。
1. カーネルゾーンホストで、管理者になります。

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Overview of Rights Profiles for Zone Administrators](#)」を参照してください。

2. **zonecfg ユーティリティーを起動し、新しいカーネルゾーン構成を作成します。**

solaris-kz のデフォルトのゾーンテンプレートは SYSsolaris-kz です。

```
global$ pfbash zonecfg -z kzone
Use 'create' to begin configuring a new zone.
zonecfg:kzone> create -t SYSsolaris-kz
```

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、[22 ページの「カーネルゾーンのリソースの管理について」](#)を参照してください。

4. **ゾーン構成を確定します。**

```
zonecfg:kzone> commit
```

5. **zonecfg ユーティリティーを終了します。**

```
zonecfg:kzone> 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。

```
global$ zoneadm -z kzone 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。

次の手順 カーネルゾーンの検証が完了したら、カーネルゾーンをインストールしてブートします。[43 ページの「カーネルゾーンのインストール」](#)を参照してください。

カーネルゾーンのリソースの管理について

ゾーン構成リソースを使用すると、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にのみ適用されます。

このセクションでは、次のコンポーネントのリソースを管理する方法について説明します。

- [23 ページの「カーネルゾーンの CPU の管理」](#)

- 25 ページの「カーネルゾーンメモリーの管理」
- 28 ページの「カーネルゾーンストレージデバイスとブート順序の管理」
- 29 ページの「カーネルゾーンネットワークデバイスと構成の管理」
- 31 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」
- 36 ページの「suspend リソースタイプの構成」
- 37 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」
- 38 ページの「IPoIB とカーネルゾーンの操作」

カーネルゾーンのリソースを管理するには、大域ゾーンで `zonecfg` コマンドを使用します。

注記 - `zonecfg` コマンドを使用するには、大域ゾーンで適切なゾーン管理権利が割り当てられた管理者である必要があります。詳細については、『[Creating and Using Oracle Solaris Zones](#)』の「[Overview of Rights Profiles for Zone Administrators](#)」を参照してください。

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(7)` 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

カーネルゾーンには、作成時にデフォルトで 4 つの仮想 CPU が割り当てられます。仮想 CPU の数を変更するには、次のいずれかの方法を使用します。

- `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` 近傍性グループに含まれる CPU の追加

カーネルゾーンへの `dedicated-cpu` リソースの追加

最良のパフォーマンスを得るには、`dedicated-cpu` リソースを構成してください。この値を設定すると、カーネルゾーンは選択された CPU 上でのみ実行されるようになります。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。システムに関する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` の出力は、システム `global` には 6 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 6 cores and 48 virtual processors (0-47)
The core has 8 virtual processors (0-7)
The core has 8 virtual processors (8-15)
The core has 8 virtual processors (16-23)
The core has 8 virtual processors (24-31)
The core has 8 virtual processors (32-39)
The core has 8 virtual processors (40-47)
SPARC-T4 (chipid 0, clock 2998 MHz)
```

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。整合性のある結果を得るには、例5「[カーネルゾーンへの専用 CPU の割り当て](#)」を参照してください。

例 5 カーネルゾーンへの専用 CPU の割り当て

この例では、`kzone1` に割り当てられた CPU や専用の CPU が存在しないことを確認したあとで CPU を割り当てる方法を示しています。`kzone1` は、6 コアの SPARC T4 が稼働するグローバルシステム上に存在していると仮定されています。

```
global$ pfbash zonecfg -z kzone1
zonecfg:kzone1> info dedicated-cpu
zonecfg:kzone1> add dedicated-cpu
zonecfg:kzone1:dedicated-cpu> set cpus=8-15
zonecfg:kzone1:dedicated-cpu> end
zonecfg:kzone1> info dedicated-cpu
      cpus: 8-15
zonecfg:kzone1:dedicated-cpu> end
zonecfg:kzone1> exit
```

詳細については、『[Oracle Solaris Zones Configuration Resources](#)』の「[dedicated-cpu Resource Type](#)」を参照してください。

カーネルゾーンへの `virtual-cpu` リソースの追加

`virtual-cpu` リソースタイプは、カーネルゾーンで可視になる仮想 CPU の数を指定します。ホストシステム上では、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を与えることがあります。詳細情報や例については、『[Oracle Solaris Zones Configuration Resources](#)』の「[virtual-cpu Resource Type \(solaris-kz Only\)](#)」を参照してください。

近傍性グループに含まれる CPU のカーネルゾーンへの追加

近傍性グループに含まれる CPU を指定できます。近傍性グループがベースとなるネットワークデバイスと同じである場合、近傍性グループに含まれる CPU を指定すると、ネットワークのパフォーマンスが向上する可能性があります。

近傍性グループの操作の詳細については、『[Oracle Solaris 12 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、「[Creating and Managing Virtual Networks](#)」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` リソースタイプの `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。カーネルゾーンのブートが完了すると、`capped-memory` リソースに指定されたメモリーがすべて、ホストのオペレーティングシステムやほかのゾーンから利用できなくなります。

カーネルゾーンの CPU とメモリーのデフォルト構成は4つの VCPU と 4G バイトのメモリーになっていますが、これは、カーネルゾーン内でアプリケーションを実行できるようにするためです。より大きなワークロードを管理するには、カーネルゾーンのメモリーサイズ (`capped-memory:physical`) を増やしてください。

- SPARC ベースシステムでは 256M バイトの増分でリソースを設定します。
- x86 ベースシステムでは 2M バイトの増分でリソースを設定します。

`capped-memory:pagesize-policy` プロパティは、カーネルゾーンの物理メモリー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、[26 ページの「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」](#)を参照してください。

追加のカーネルゾーンテンプレート `SYSsolaris-kz-minimal` は、サポートされている最小のカーネルゾーン構成である1つの VCPU と 2G バイトのメモリーを提供します。

注記 - Fujitsu SPARC M12 サーバー、Fujitsu M10 サーバー、または SPARC M10 サーバーでは、`SYSsolaris-kz-minimal` テンプレートで作成されたカーネルゾーンはメモリー不足のため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ーを増やしてください。

`capped-memory` ゾーンリソースタイプの設定の詳細については、『[Oracle Solaris Zones Configuration Resources](#)』の「[Capped Memory Guidelines for a solaris-kz Zone](#)」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ー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は、`zvo1` 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、`zvo1` のサイズは 16G バイトです。異なるルートディスクサイズを指定するには、`zoneadm install -x install-size` コマンドを使用します。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、インストール時に次のコマンドを使用します。

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

`zoneadm` コマンドを使用してディスクサイズを変更する方法については、[zoneadm\(8\)](#) のマニュアルページを参照してください。

例 6 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 ベースシステム上のカーネルゾーンで 2048M バイトのメモリーを指定する方法を示しています。

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

例 7 x86 ベースシステムでの capped-memory リソースの設定

この例では、x86 ベースシステム上のカーネルゾーンで 16G バイトのメモリーを指定する方法を示しています。

```
global$ pfbash zonecfg -z kzone1
zonecfg:kzone1> select capped-memory
zonecfg:kzone1:capped-memory> set physical=16g
zonecfg:kzone1:capped-memory> end ; 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を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。

カーネルゾーンの `pagesize-policy` プロパティー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。このページサイズは適切でない可能性があります。例8「最大ページサイズが使用されるように物理メモリーを設定する」で示すように、サポートされる最大のページサイズの整数倍となる量に、`physical` プロパティーを設定する必要があります。

割り当てるメモリーの量は、必要なページサイズと完全に整合性が取れている必要があります。したがって、次のいずれかの条件に該当する場合は `pagesize-policy` をクリアする必要があります。

- ターゲットシステムのページサイズがソースシステムより小さい場合。
- ソースのカーネルゾーンが Oracle Solaris 11.4 の更新版で作成されたものであり、ターゲットが Oracle Solaris 11.3 の初回リリースなど、`pagesize-policy` プロパティーをサポートしない Oracle Solaris リリースである場合。

例9「以前の Oracle Solaris リリースにカーネルゾーンを移行する前に `pagesize-policy` プロパティーをクリアする」を参照してください。

例 8 最大ページサイズが使用されるように物理メモリーを設定する

次の出力からわかるように、SPARC T5 ではさまざまなページサイズがサポートされています。最大のものは 2147483648 バイト、つまり 2G バイトです。

2G バイトのページサイズを増分として使用し、`capped-memory:physical` プロパティーが 8G バイトに設定されています。`pagesize-policy` が `largest-available` に設定されている場合、この値を指定すると、システムは最大のページサイズを使用できます。

```
global$ pfbash pagesize -a
8192
65536
4194304
268435456
2147483648
global$ pfbash 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 ; exit
global$ reboot
```

例 9 以前の Oracle Solaris リリースにカーネルゾーンを移行する前に `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することで、このプロパティをサポートしていない Oracle Solaris リリース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$ pfbash -z kzone1
zonecfg:kzone1> select capped-memory
zonecfg:kzone1:capped-memory> clear pagesize-policy
zonecfg:kzone1:capped-memory> end
zonecfg:kzone1> commit
global$ reboot
```

`zoneadm migrate -n` コマンドを使って移行の試験実行を行うと、`pagesize-policy` をクリアする必要があるかどうかわかります。詳細は、70 ページの「[zoneadm migrate コマンドについて](#)」を参照してください。

ヒント - パフォーマンスが最大ページサイズの使用に依存するデータベースのようなアプリケーションをカーネルゾーンがホストする場合、最大ページサイズが使用されないかぎりブートが実行されないように、`pagesize-policy=largest-only` を設定することをお勧めします。

`physical` および `pagesize-policy` プロパティの設定の詳細については、『[Oracle Solaris Zones Configuration Resources](#)』の「[Capped Memory Guidelines for a solaris-kz Zone](#)」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あり、デフォルトでは 16G バイトの ZFS ボリュームになります。ディスク領域を増やすには、25 ページの「[カーネルゾーンのメモリの管理](#)」の説明に従ってルートディスクを大きくします。または、ストレージデバイスを追加します。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/dsk/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有効なストレージ URI によって定義する必要があります。
- 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。

各ストレージデバイスのブート順序を指定するには、`device` リソースタイプの `bootpri` プロパティを使用します。`bootpri` プロパティは任意の正の整数値に設定します。



注意 - ブートデバイス以外のデバイスで `bootpri` プロパティを設定してはいけません。`bootpri` 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` プロパティの設定を解除するには、`zonecfg clear bootpri` コマンドを使用します。

各デバイスのデフォルトのブート順は、まず `bootpri` プロパティでデバイスをソートし、複数のデバイスが同じ `bootpri` 値を持つ場合はさらに `id` プロパティでソートすることによって決定されます。

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。

例 10 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/dsk/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$ pfbash zonecfg -z kzone1
zonecfg:kzone1> add device
zonecfg:kzone1:device> set storage=dev:/dev/dsk/c9t0d0
zonecfg:kzone1:device> set bootpri=4
zonecfg:kzone1:device> end
```

例 11 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$ pfbash 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:
  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 2 つのリソースタイプの詳細について

は、『Oracle Solaris Zones Configuration Resources』の「Configurable Resource Types and Global Properties」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がベストプラクティスです。

カーネルゾーンは排他的 IP ゾーンである必要があります。排他的 IP ゾーンについては、『Creating and Using Oracle Solaris Zones』の「Exclusive-IP Zone Network Address」を参照してください。

追加の MAC アドレスを指定すれば、カーネルゾーン内でのネイティブ (`solaris`) ゾーンの実行をサポートできます。詳細は、102 ページの「カーネルゾーン内の非大域ゾーンの管理」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。

例 12 カーネルゾーンへのネットワークデバイスの追加

この例は、ネットワーク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。ID 3 は、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を指定します。ゾーンのブート後に `dladm show-phys -i` コマンドを実行すると、カーネルゾーン内に暗黙的に作成された物理リンクに関する情報が表示されます。ID 列の値は、`zonecfg` で設定された ID と一致します。

```
global$ pfbash zonecfg -z kzone1
zonecfg:kzone1> add anet
zonecfg:kzone1:anet> set id=3
zonecfg:kzone1:anet> info
anet 1:
      id: 3
zonecfg:kzone1:anet> end ; exit

global$ zoneadm -z kzone1 boot
global$ zlogin kzone1 dladm show-phys -i
LINK           MEDIA      ID          DEVICE      ACTIVE      STANDBY
net0            Ethernet  anet:0     vnic1000    --          zvnet0
net1            Ethernet  anet:3     vnic1001    --          zvnet1
```

例 13 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、ID 1 の `anet` デバイスが削除されます。

```
global$ pfbash zonecfg -z kzone1 info anet
```

```
anet:
  configure-allowed-address: true
  id: 0
anet:
  configure-allowed-address: true
  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 anet リソースタイプの iov プロパティを使用すると、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、ベアメタル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にシステムハードウェアに実装されています。

カーネルゾーンの anet リソースで SR-IOV を有効化するには、大域ゾーンのデータリンク上で SR-IOV を有効にする必要があります。Oracle Solaris での SR-IOV の使用方法については、『[Oracle Solaris 12 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「[Using Single Root I/O Virtualization With VNICs](#)」を参照してください。

iov プロパティは、カーネルゾーンとネイティブ (solaris) ゾーンでのみサポートされます。

iov プロパティを有効にすると、カーネルゾーンを一時停止および再開し、ウォーム移行またはライブ移行を使用してカーネルゾーンを移行する機能が、Oracle Solaris 11.4 を実行しているホストシステムとゾーンに限定されます。詳細は、[34 ページの「SR-IOV が有効になった anet リソースを含むカーネルゾーンの移行について」](#)を参照してください。

anet リソースタイプの iov プロパティを有効化および構成する方法については、『[Oracle Solaris Zones Configuration Resources](#)』の「[Zone Global Properties](#)」を参照してください。

ヒント - SR-IOV をサポートする一部の Intel ネットワークアダプタを使用している場合、仮想機能が悪意のある動作のターゲットとなる可能性があります。ソフトウェアにより生成された予期しないフレームによってホストシステムと仮想スイッチ間のトラフィックが減速し、パフォーマンスに悪影響を与えることがあります。この問題を回避するには、SR-IOV が有効になっているすべてのポートが VLAN タグ付けを使用するように構成することで、予期しない悪質な可能性のあるフレームがドロップされるようになります。詳細は、[例15「anet リソースでの SR-IOV および VLAN タグ付けの構成」](#)を参照してください。

▼ 単一の anet リソース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. ゾーン管理者になります。

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」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、選択された anet リソースの iov プロパティを設定します。

```
$ pfbash zonecfg -z kernel-zone
zonecfg:kernel-zone> select anet id=id-number
zonecfg:kernel-zone:anet> set lower-link=network-interface
zonecfg:kernel-zone>set iov=iov-value
zonecfg:kernel-zone:anet> set iov=auto
zonecfg:kernel-zone:anet> end ; exit
```

次の例では、カーネルゾーン kzone1 に属する anet リソースで iov プロパティを有効化する方法を示しています。

```
global$ pfbash 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 ; exit
```

3. (オプション) カーネルゾーン構成の anet リソースに 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 上のカーネルゾーン kzone1 の、ID 0 を持つ anet リソースの場合:

```
$ zonecfg -z kzone1 info anet id=0
anet:
  lower-link: net1
  configure-allowed-address: true
  iov: auto
  id: 0
```

4. 選択されたネットワークインタフェースで 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 -i
```

このコマンドの出力は、ホストシステムの大域ゾーンとカーネルゾーンで実行されている Oracle Solaris のバージョンに応じて変わります。選択された Oracle Solaris バージョンの組み合わせに対するサンプル出力を次に示します。

- 大域ゾーンとカーネルゾーンの両方で Oracle Solaris 11.4 が実行されている:

```
global$ pfexec zlogin kzone
kzone$ dladm show-phys -i
```

LINK	MEDIA	ID	DEVICE	ACTIVE	STANDBY
net0	Ethernet	anet:0	vnic1000	ixgbev0	zvnet0

- 大域ゾーンで Oracle Solaris 11.4 が、カーネルゾーンで Oracle Solaris 11.3 が実行されている:

```
global$ pfexec zlogin kzone
kzone$ dladm show-phys -i
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	down	0	unknown	ixgbev0

- 大域ゾーンで Oracle Solaris 11.3 が、カーネルゾーンで Oracle Solaris 11.4 が実行されている:

```
global$ pfexec zlogin kzone
kzone$ dladm show-phys -i
```

LINK	MEDIA	ID	DEVICE	ACTIVE	STANDBY
net0	Ethernet	anet:0	vnic1000	ixgbev0	--

例 14 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 `auto` に設定されています。値がデフォルト値の `off` に設定されている場合、値は表示されません。

```
global$ pfbash zonecfg -z kzone1
zonecfg:kzone1> select anet id=0
zonecfg:kzone1:anet> info
anet:
  lower-link: net1
  configure-allowed-address: true
  iov: auto
  id: 0
```

例 15 anet リソースでの SR-IOV および VLAN タグ付けの構成

次の例は、anet リソース上で VLAN タグ付けが有効になるよう 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$ pfbash 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 ; exit
```

VLAN ID の設定や VLAN タグ付けの詳細については、[38 ページの「カーネルゾーンでの仮想 LAN の構成」](#)を参照してください。

SR-IOV が有効になった anet リソースを含むカーネルゾーンの移行について

SR-IOV を使用しているカーネルゾーンの場合、カーネルゾーンを一時停止および再開し、ウォーム移行またはライブ移行を使用して移行を行う機能が、Oracle Solaris 11.4 を実行しているホストシステムとゾーンに限定されます。カーネルゾーンの構成に設定 `iov=auto` または `iov=on` が含まれている場合、ソースホスト、ターゲットホスト、カーネルゾーンのいずれかで古いリリースが実行されていると、移行が失敗します。

`iov` が有効になったカーネルゾーンを移行する必要がある、かつカーネルゾーン、ソースホストの大域ゾーン、ターゲットホストの大域ゾーンのいずれかで Oracle Solaris 11.4 より古いリリースが実行されている場合は、コールド移行を実行する必要があります。

移行や `iov` プロパティの制限の詳細については、次を確認してください。

- [第5章「Oracle Solaris カーネルゾーンの移行」](#)
- 『[Oracle Solaris Zones Configuration Resources](#)』 の「[Zone Global Properties](#)」

SR-IOV が有効になったカーネルゾーンでのネットワーク高可用性の構成

SR-IOV が有効になったデータリンクマルチパス (DLMP) が大域ゾーンで使用されている場合、カーネルゾーン内の SR-IOV デバイスでネットワーク高可用性を実現できます。

高可用性を有効にするには、大域ゾーンで SR-IOV が有効になったリンクの DLMP リンクアグリゲーションを作成したあと、その DLMP アグリゲーションを下位リンクと

して使用する anet リソースをカーネルゾーンに追加します。anet リソースの iov プロパティを auto に設定し、ゾーンをブートします。

詳細は、『[Managing Network Datalinks in Oracle Solaris 11.4](#)』の「[Datalink Multipathing Aggregations](#)」を参照してください。

例 16 カーネルゾーンでのネットワーク高可用性 DLMP リンクアグリゲーションの構成

```
global$ pfbash dladm set-linkprop -p iov=on net3
global$ dladm set-linkprop -p iov=on net4
global$ dladm create-aggr -l net3 -l net4 -m dlmp dlmp0
global$ zonecfg -z kz1
zonecfg:kz1> create -t SYSsolaris-kz
zonecfg:kz1> add anet
zonecfg:kz1:anet> set lower-link=dlmp0
zonecfg:kz1:anet> set iov=auto
zonecfg:kz1:anet> set id=0
zonecfg:kz1:anet> end

global$ zoneadm -z kz1 boot
global$ dladm show-aggr -c dlmp0
LINK      PORT      SPEED DUPLEX  STATE  CLIENTS
dlmp0     --        10000Mb full  up     --
          net3      10000Mb full  up     kz1/net0
          net4      10000Mb full  up     dlmp0

global$ zlogin kz1 dladm show-phys -i
LINK      MEDIA      ID      DEVICE      ACTIVE      STANDBY
net0      Ethernet   anet:0  vnic1000    ixgbevfo    zvnet0
```

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、ホストシステム内のシャドウ 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
```

VF は DLMP アグリゲーションから割り当てることができます。DLMP アグリゲーションに `iovs=auto` を設定できます。この場合、利用可能な VF リソースが存在すると VF が割り当てられます。例については、[例16「カーネルゾーンでのネットワーク高可用性 DLMP リンクアグリゲーションの構成」](#) を参照してください。

DLMP またはトランクアグリゲーションで `iovs=on` を設定することは、禁止されています。

`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。追加情報については、[dladm\(8\)](#) および [zonecfg\(8\)](#) 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追加情報については、『[Oracle Solaris 12 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。

suspend リソースタイプの構成

カーネルゾーンの構成に `suspend` リソースがある場合のみ、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。一時停止されたイメージ用のストレージデバイスは、`capped-memory:physical` プロパティ経由でカーネルゾーンに割り当てられたメモリーの量に十分対応できるだけの大きさでなければいけません。

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` グローバル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` グローバルプロパティの詳細については、[zonecfg\(8\)](#) を参照してください。

注記 - カーネルゾーンが SR-IOV を使用している場合、カーネルゾーンを一時停止してウォーム移行またはライブ移行する機能が、Oracle Solaris 11.4 を実行しているホストシステムとゾーンに限定されます。詳細は、[34 ページの「SR-IOV が有効になった anet リソースを含むカーネルゾーンの移行について」](#) を参照してください。

例 17 suspend リソースを構成してカーネルゾーンの一時停止を有効にする

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$ pfbash 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
```

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 18 ウォーム移行が可能となるように 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$ pfbash 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 ; exit
```

詳細は、[80 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 リソースタイプの要件の詳細については、[solaris-kz\(7\)](#) 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。検証済みブートは、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` リソースタイプを使用します。ベリファイドブートと `verified-boot` リソースタイプは、`solaris-kz` ブランドゾーンでのみサポートされます。プロパティの例や情報については、『[Oracle Solaris Zones Configuration Resources](#)』の「[verified-boot Resource Type](#)」を参照してください。

証明書の検証やベリファイドブートの追加情報については、`elfsign(1)` のマニュアルページおよび『[Oracle Solaris 11.4 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保護](#)』の「[ベリファイドブートの使用](#)」を参照してください。

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` リソースタイプのプロパティを設定します。『[Oracle Solaris Zones Configuration Resources](#)』の「[Zone Global Properties](#)」および『[Oracle Solaris 12 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「[Creating and Viewing Paravirtualized IPoIB Datalinks in Kernel Zones](#)」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` リソースを使用すれば、カーネルゾーンの内に `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` リソースタイプの詳細については、『[Oracle Solaris Zones Configuration Resources](#)』の「[Configurable Resource Types and Global Properties](#)」を参照してください。
- `vlan` リソースタイプは、カーネルゾーンを VLAN 対応のものにします。ホストシステムは、これらの VLAN 宛てのパケットを、VLAN タグを取り除かないままカーネルゾーンに転送します。その後、カーネルゾーン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 19 追加の VLAN を使用したカーネルゾーンの構成

mac-address 0:1:2:3:4:5、PVID 11、および 2 つの追加の VID 45 と 46 を使用して、kzone0 を構成します。

```
global$ pfbash zonecfg -z kzone0
zonecfg:kzone0> create -t SYSsolaris-kz
zonecfg:kzone0> select anet id=0
zonecfg:kzone0> set mac-address=0:1:2:3:4:5
zonecfg:kzone0:anet> set vlan-id=11
zonecfg:kzone0:anet> add vlan
zonecfg:kzone0:anet:vlan> set vlan-id=45
zonecfg:kzone0:anet:vlan> end
zonecfg:kzone0:anet> add vlan
zonecfg:kzone0:anet:vlan> set vlan-id=46
zonecfg:kzone0:anet:vlan> end
zonecfg:kzone0:anet> info vlan
  vlan 0:
    vlan-id: 45
  vlan 1:
    vlan-id: 46
zonecfg:kzone0:anet> end
zonecfg:kzone0> commit ; 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の出力が表示されます。

```
global$ dladm show-vnic
LINK          OVER    SPEED  MACADDRESS  MACADDRTYPE  IDS
kzone0/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 タプ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 タプ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 kzone0 に渡されるので、kzone0 で VID 11 タグが認識され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 46 のタグが付いたまま kzone0 に渡されます。

VID が 45 の VLAN データリンク vlan45 が kzone0 内にある場合、kzone0 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。kzone0

の `vlan45` データリンクから送出されたフレームはすべて、`kzone0` 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 20 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で、`dladm show-phys -v` コマンドを使用して、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 kzone0
kzone0$ 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、クラウド配備などの場合は、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。そのような場合は次の 2 つの構成オプションがあります。

- 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が可能です。

ヒント - 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してください。静的構成は、SR-IOV VF ベースの `anet` リソースでも必要になります。

動的構成を有効にするには、次の手順に示すように、`anet` リソースタイプの `allowed-mac-address` および `allowed-vlan-ids` を設定します。

これらのプロパティの詳細については、『[Oracle Solaris Zones Configuration Resources](#)』の「[anet Resource Type](#)」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. ゾーン管理者になります。

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」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。

```
$ pfbash 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 allowed-vlan-ids 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 ; 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. カーネルゾーン内で、許可されるアドレスと ID を確認します。

許可されている MAC 接頭辞と VLAN ID を確認するには、dladm show-phys コマンドで、-o オプションを使用して出力フィールドを指定します。たとえば、kzone1 で確認する場合:

```
global$ zlogin kzone1
kzone1$ dladm show-phys -o link,media,id,allowed-addresses,allowed-vids
LINK          MEDIA          ID             ALLOWED-ADDRESSES  ALLOWED-VIDS
net0          Ethernet       anet:0         fa:16:3f,          100-199,
                                     fa:80:20:21:22    400-498,500
```


Oracle Solaris カーネルゾーンのインストール、シャットダウン、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。

- 43 ページの「カーネルゾーンのインストール」
- 53 ページの「カーネルゾーンのアンインストール」
- 53 ページの「カーネルゾーンのシャットダウン、停止、およびリブート」
- 54 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、『[Introduction to Oracle Solaris Zones](#)』の「[How Non-Global Zones Are Created](#)」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2章「Oracle Solaris カーネルゾーンの構成」](#)の説明に従ってゾーンを構成する必要があります。構成が完了したら、ゾーンの適切なインストール方法を選択できます。

- 大域ゾーンの構成と同一のカーネルゾーンを1つ作成する – [44 ページの「カーネルゾーンの直接インストール」](#)
- 同一の要件を持つカーネルゾーンを複数作成する – [46 ページの「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」](#)
- 同一の要件を持つカーネルゾーンを複数作成する – [49 ページの「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」](#)
- 既存のゾーンを複製するか、既存のゾーンを複製して変更する – [51 ページの「インストールイメージを使用したカーネルゾーンのインストール」](#)
- システムから既存のインストール済みのゾーンをコピーし、そのコピーをシステムに追加する – [54 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中、2つの場所にログが記録されます。ログの内容は同じです。

- 大域ゾーン内の /var/log/zones/ ディレクトリ内
- インストール対象ゾーン内の /zones/zonename/root/var/log/zones/ ディレクトリ内

カーネルゾーンの直接インストール

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。直接インストールは、カーネルゾーンのもっとも単純な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。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、大域ゾーン内で実行されている Oracle Solaris とまったく同じバージョンをインストールします。異なるバージョンの Oracle Solaris をインストールするには、別のインストール方法を選択する必要があります。

▼ カーネルゾーンの直接インストールを行う方法

1. カーネルゾーンホストで、管理者になります。

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Overview of Rights Profiles for Zone Administrators](#)」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。

```
global$ pfbash zoneadm -z kzone 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 12 パッケージリポジトリのコピーと作成](#)』の「[Troubleshooting Local Package Repositories](#)」を参照してください。

3. (オプション) インストールログを確認します。

それらの場所については、43 ページの「カーネルゾーンのインストール」を参照してください。

4. カーネルゾーンをブートします。

```
global$ zoneadm -z kzone boot
```

これでゾーンにログインできます。Oracle Solaris にはいくつかのログイン方法が用意されています。カーネルゾーンの方法は非大域ゾーンの方法と同じです。詳細は、『[Creating and Using Oracle Solaris Zones](#)』の第7章、「[Logging In to Non-Global Zones](#)」を参照してください。

例 21 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$ pfbash 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.
```

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

大域ゾーンとは異なる特定のリソースおよびパッケージ構成を持つカーネルゾーンを複数インストールする必要がある場合は、Automated Installation (AI) マニフェストを使用できます。

代替 AI マニフェストをカーネルゾーンにインストールする場合は、次の要件やガイドラインに従ってください。

- 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。99 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。
- 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、51 ページの「[インストールイメージ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。

▼ AI マニフェストを使用してカーネルゾーンをインストールする方法

この手順では、AI マニフェストを使用してカーネルゾーンをインストールするための `zoneadm` コマンドと詳細な例を示します。

AI マニフェストの開発やカスタマイズに関する追加情報については、『[Customizing Automated Installations With Manifests and Profiles](#)』の第5章、「[Specifying Criteria for AI Manifests and System Configuration Profiles](#)」を参照してください。

ゾーンのルートディスクの作成方法の詳細については、『[Creating and Using Oracle Solaris Zones](#)』の「[Zone Installation and Administration Concepts](#)」を参照してください。

始める前に 46 ページの「[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](#)」の要件やガイドラインを確認します。

1. **カーネルゾーンホストで、管理者になります。**
詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Overview of Rights Profiles for Zone Administrators](#)」を参照してください。

2. **指定された AI マニフェスト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 kzone install -m path-to-manifest
```

3. **(オプション) インストールログを確認します。**

それらの場所については、43 ページの「[カーネルゾーンのインストール](#)」を参照してください。

例 22 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$ pfbash 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

```

例 23 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 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>
  </ai_instance>
</auto_install>

```



```

</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
```

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

大域ゾーンとは異なる特定のリソースおよびパッケージ構成を持つカーネルゾーンを複数インストールする必要がある場合は、sysconfig プロファイルを使用できます。

▼ sysconfig プロファイルを使用してカーネルゾーンをインストールする方法

この手順では、zoneadm コマンドで sysconfig プロファイルを使用してカーネルゾーンをインストールする方法の詳細な例を示します。

始める前に sysconfig プロファイルを作成し、カーネルゾーンのインストールに使用するシステムからアクセスできる場所にそれを配置します。詳細は、『[Oracle Solaris 12 システムの自動インストール](#)』の第9章、「[Unconfiguring or Reconfiguring an Oracle Solaris Instance](#)」、および [sysconfig\(8\)](#) と [solaris-kz\(7\)](#) のマニュアルページを参照してください。

sysconfig プロファイルを使用してカーネルゾーンをインストールする方法の詳細については、[zoneadm\(8\)](#) のマニュアルページを参照してください。

1. カーネルゾーンホストで、管理者になります。
詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Overview of Rights Profiles for Zone Administrators](#)」を参照してください。
2. sysconfig プロファイルを使用してカーネルゾーンをインストールします。

```
global$ pfbash zoneadm -z kzone install -c sysconfig-profile
```

3. (オプション) インストールログを確認します。

それらの場所については、[43 ページの「カーネルゾーンのインストール」](#)を参照してください。

例 24 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$ pfbash 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
```

インストールイメージを使用した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イメージのみからインストールすることも、特定のリソースおよびパッケージ構成を含む AI マニフェストとインストールイメージの組み合わせからインストールすることもできます。

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場合は、次の要件やガイドラインに従ってください。

- 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。16 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。
- ISO イメージへの完全なパスを指定する必要があります。
- 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はサポートされていません。詳細については、次を参照してください。
 - 『[Manually Installing an Oracle Solaris 11.4 System](#)』の第3章、「[Using the Text Installer](#)」
 - 『[Automatically Installing Oracle Solaris 11.4 Systems](#)』の第7章、「[Automated Installations That Boot From Media](#)」

このセクションでは、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ための次の手順について説明します。

- 51 ページの「[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](#)」
- 52 ページの「[インストールイメージと AI マニフェストからカーネルイメージをインストールする方法](#)」

▼ 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

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、『[Creating and Using Oracle Solaris Zones](#)』の「[Zone Console Login](#)」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または切断すると、インストールが失敗します。

- 始める前に
- [51 ページの「インストールイメージを使用したカーネルゾーンのインストール」](#)の要件やガイドラインを確認します。
 - カーネルゾーンのインストールに使用するシステムからインストールイメージにアクセスできることを確認します。

1. **カーネルゾーンホストで、管理者になります。**

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Overview of Rights Profiles for Zone Administrators](#)」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$ pfbash zoneadm -z kzone install -b path-to-iso-file
```

3. **(オプション) インストールログを確認します。**

それらの場所については、[43 ページの「カーネルゾーンのインストール」](#)を参照してください。

例 25 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

この例では、`/var/tmp/solaris-media.iso`にあるイメージをカーネルゾーン `kzone2` にインストールしています。

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

▼ インストールイメージと AI マニフェストからカーネルイメージをインストールする方法

特定のリソースやパッケージの構成情報を提供する必要がない場合は、[51 ページの「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」](#)を代わりに参照してください。

- 始める前に
- [51 ページの「インストールイメージを使用したカーネルゾーンのインストール」](#)の要件やガイドラインを確認します。
 - カーネルゾーンのインストールに使用するシステムからインストールイメージと AI マニフェストにアクセスできることを確認します。

1. ゾーン管理者になります。
詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」を参照してください。
2. インストールイメージの ISO ファイルと AI マニフェストを使用して、カーネルゾーンをインストールします。

```
global$ pfbash zoneadm -z kzone install -b path-to-iso-file -m path-to-manifest
```


zoneadm install のオプションの詳細については、[zoneadm\(8\)](#) のマニュアルページを参照してください。
3. (オプション) インストールログを確認します。
それらの場所については、[43 ページ](#)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 26 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と AI マニフェストの使用

この例では、`/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、AI マニフェスト `/var/tmp/kz_manifest.xml` を使用して特定のリソースおよびパッケージ構成を提供しています。

```
global$ pfbash 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 `running` 状態にはできません。詳細は、『[Creating and Using Oracle Solaris Zones](#)』の第2章、「[About Installing, Shutting Down, Halting, Uninstalling, and Cloning Non-Global Zones](#)」を参照してください。

カーネルゾーンのシャットダウン、停止、およびリブート

カーネルゾーンをシャットダウン、停止、およびリブートするには、`zoneadm shutdown`、`zoneadm halt`、および `zoneadm reboot` コマンドを使用します。詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[About Shutting Down, Halting, Rebooting, and Uninstalling Zones](#)」を参照してください。

ヒント - ホストシステムのリブート時にゾーンが自動的にブートされるようにするには、『[Creating and Using Oracle Solaris Zones](#)』の「[Automatic Zone Boot](#)」を確認してください。

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、ソースゾーン上の追加されたパッケージ、変更されたゾーンリソース、およびファイルの変更が、各クローン内に存在しています。ゾーンのクローニングは、似た要件を持つ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27「[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。
- 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28「[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。

詳細は、『[Oracle Solaris 12 でのシステム復旧とクローン](#)』の第2章、「[Working With Unified Archives](#)」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 27 `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。

次の例は、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、『[Creating and Using Oracle Solaris Zones](#)』の「[How to Clone a Non-Global Zone on the Same System](#)」を参照してください。

```
global$ pfbash 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 28 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 12 でのシステム復旧とクローン](#)』を参照してください。

```

global$ pfbash 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

```


カーネルゾーンのライブゾーン再構成

カーネルゾーンの実行中にそのゾーンのライブ構成を再構成したり、そのゾーンのライブ構成に関して報告したりするには、ライブゾーン再構成を使用します。

この章で扱う内容は、次のとおりです。

- 57 ページの「ライブゾーン再構成の実行」

solaris ゾーンとカーネルゾーンのライブゾーン再構成でサポートされているリソースおよびプロパティを示す表を確認するには、『Oracle Solaris ゾーン構成リソース』の「Live Zone Reconfiguration Support or Restriction for Resource Types and Global Properties」を参照してください。

ライブゾーン再構成の実行

このセクションでは、一般的なライブゾーン再構成タスクを実行するための次の手順について説明します。

- 57 ページの「実行中のゾーンのライブ構成を検査する方法」
- 58 ページの「ライブゾーン構成の影響を確認する方法」
- 58 ページの「ライブゾーンに永続的な構成変更を加える方法」
- 60 ページの「実行中のゾーンに一時的な変更を加える方法」
- 61 ページの「ライブゾーン構成の変更の確定中に障害から回復する方法」

▼ 実行中のゾーンのライブ構成を検査する方法

実行中のゾーンの構成を表示およびエクスポートするには、この手順を実行します。

1. カーネルゾーンホストで、管理者になります。

詳細は、『Creating and Using Oracle Solaris Zones』の「Overview of Rights Profiles for Zone Administrators」を参照してください。

2. ライブゾーン構成に関する情報を表示します。

```
global$ pfbash zonecfg -z kzone -r info
```

3. (オプション) ライブ構成をエクスポートします。

```
global$ zonecfg -z kzone -r export -f exported.cfg
```

▼ ライブゾーン構成の影響を確認する方法

ライブゾーン構成に加えた変更の最終確定を行う前に、それらの変更を確認するには、この手順を実行します。

始める前に 次のドキュメントを確認してください。

- 『Creating and Using Oracle Solaris Zones』 の 「Live Zone Reconfiguration Dry Run」
- 『Oracle Solaris Zones Configuration Resources』 の 付録 A, 「Resource Types and Global Properties That Support Live Zone Reconfiguration,」

1. カーネルゾーンホストで、管理者になります。

詳細は、『Creating and Using Oracle Solaris Zones』 の 「Overview of Rights Profiles for Zone Administrators」 を参照してください。

2. ライブモードで zonecfg ユーティリティを起動し、目的のゾーン変更を構成します。

```
global$ pfbash zonecfg -z kzone -r
zonecfg:kzone>          ゾーン構成の変更を行います
```

3. 再構成によって実行されるアクションを表示します。

-n オプションを指定すると、ゾーンの変更が実際に確定されるのを防げます。

```
zonecfg:kzone> commit -n
```

次の手順 確認した変更をライブゾーン構成に加える場合は、-n オプションを指定せずに zonecfg commit コマンドを発行します。

▼ ライブゾーンに永続的な構成変更を加える方法

ゾーンのリブート後も持続するライブゾーン構成の変更を行うには、この手順を実行します。

始める前に 次のドキュメントを確認してください。

- 『Creating and Using Oracle Solaris Zones』 の 「zonecfg Utility Edit Modes」

- 『Creating and Using Oracle Solaris Zones』 の 「Persistent Changes to the Live Zone Configuration」
- 『Oracle Solaris Zones Configuration Resources』 の 付録 A, 「Resource Types and Global Properties That Support Live Zone Reconfiguration,」

1. カーネルゾーンホストで、管理者になります。

詳細は、『Creating and Using Oracle Solaris Zones』 の 「Overview of Rights Profiles for Zone Administrators」 を参照してください。

2. デフォルトモードでゾーンに変更を加えます。

フォーマットについては、『Making Persistent Changes to the Live Configuration』 『Creating and Using Oracle Solaris Zones』 の および、zonecfg(8) のマニュアルページを参照してください。

3. ライブ構成に変更を適用します。

```
global$ pfbash zoneadm -z kzone apply
```

例 29 ライブゾーン構成における仮想 CPU の数の削減

この例は、16 個の仮想 CPU が構成されている実行中のカーネルゾーン kz1 を示しています。VCPU の数を 8 に設定するように永続的構成が変更され、その変更がライブ構成に適用されます。出力には、カーネルゾーンでそれらの CPU のいくつかの使用を停止できない場合にどうなるかが表示されます。これは、dedicated-cpu リソースタイプで構成された solaris ゾーンがカーネルゾーンに含まれている場合などに発生することがあります。

ライブゾーン再構成では、カーネルゾーン (ゲスト) から削除できないそれらの CPU をスキップする一方、その他の CPU の削除を試みることで、この要求を満たそうとします。指定された数の CPU を削除できない場合は、部分的に操作が成功し、出力には仮想 CPU の新しい数が表示されます。

```
global$ pfbash zonecfg -z kz1 -r info virtual-cpu
virtual-cpu:
    ncpus: 16
global$ zonecfg -z kz1 'select virtual-cpu;set ncpus=8;end'
global$ zoneadm -z kz1 apply
zone 'kz1': Checking: Modifying virtual-cpu ncpus=8
zone 'kz1': Applying the changes
zone 'kz1': error: dr-cpu failed for cpu id=15: Operation was blocked
zone 'kz1': error:          status: CPU is configured for use by the guest
zone 'kz1': error: dr-cpu failed for cpu id=14: Operation was blocked
zone 'kz1': error:          status: CPU is configured for use by the guest
...
    8つの仮想CPUの削除を試みる操作が継続されます
...
zone 'kz1': warning: operation succeeded partially for virtual cpus (requested: 8, final:
12)
global$ zonecfg -z kz1 -r info virtual-cpu
```

```
virtual-cpu:  
ncpus: 12
```

▼ 実行中のゾーンに一時的な変更を加える方法

実行中のゾーンのライブ構成を一時的に変更したあと、永続的構成を復元して変更を元に戻すには、この手順を実行します。

始める前に 次のドキュメントを確認してください。

- 『Creating and Using Oracle Solaris Zones』 の 「zonecfg Utility Edit Modes」
- 『Creating and Using Oracle Solaris Zones』 の 「Temporary Changes to the Live Zone Configuration」
- 『Oracle Solaris Zones Configuration Resources』 の 付録 A, 「Resource Types and Global Properties That Support Live Zone Reconfiguration,」

1. カーネルゾーンホストで、管理者になります。

詳細は、『Creating and Using Oracle Solaris Zones』 の 「Overview of Rights Profiles for Zone Administrators」 を参照してください。

2. ゾーン構成を変更します。

サンプルコマンドでは、ライブモードでディスクが追加され、コマンド出力が表示されます。

注記 - zonecfg commit コマンドは必要ありません。zonecfg ユーティリティーは、コマンドの終了時に変更を確定します。

```
global$ pfbash zonecfg -z kzone -r 'add device;set storage=dev:/dev/dsk/cNtXd;end'  
zone 'kzone': Checking: Adding device storage=dev:/dev/dsk/cNtXd  
zone 'kzone': Applying the changes
```

3. (オプション) その構成変更が必要なくなったら、ゾーンを永続的ゾーン構成に戻します。

サンプルコマンドでは、一時的な構成変更が削除されます。

```
global$ zoneadm -z kzone apply  
zone 'kzone': Checking: Removing device storage=dev:/dev/dsk/cNtXd  
zone 'kzone': Applying the changes
```

別の方法として、ゾーンをリブートしてライブゾーン構成の変更を破棄し、永続的ゾーン構成に戻すこともできます。

注意事項 commit 操作でエラーが報告された場合は、61 ページの「ライブゾーン構成の変更の確定中に障害から回復する方法」を参照してください。

▼ ライブゾーン構成の変更の確定中に障害から回復する方法

ライブゾーン構成の編集集中に、実行中のゾーンの構成が外部で変更されることがあります。こうした競合が発生すると、`zonecfg commit` コマンドからエラーが返されます。

この手順を実行すると、ゾーン構成をリロードして更新されたバージョンを表示し、編集をやり直すことで、エラーを解消できます。

1. カーネルゾーンホストで、管理者になります。

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Overview of Rights Profiles for Zone Administrators](#)」を参照してください。

2. `reload` サブコマンドを発行し、正しい構成変更を行なって、その変更を確定します。

このステップは、ユーザーがまだ一時的な構成変更の確定に失敗した `zonecfg` のセッション中であることを前提とします。

```
zonecfg:kzone> reload
zonecfg:kzone> temporary-configuration-changes
zonecfg:kzone> commit
```

例 30 失敗した一時的なゾーン構成変更からの回復

次の例は、一時的な構成変更、変更が失敗したことを示すエラーメッセージ、回復ステップ、および変更が成功したことを確認する出力を示しています。

```
ライブモードでの構成変更の例
global$ pfbash zonecfg -z kzone1 -r
zonecfg:kzone1> add anet;set lower-link=net1;set maxbw=2G;end
zonecfg:kzone1> commit
zone 'kzone1': error: the live configuration has changed externally.
Trying to commit changes to externally changed live configuration
Please use reload to start again. Your local changes will be lost.
```

```
構成をリロードします
zonecfg:kzone1> reload
Are you sure you want to reload (y/[n])? y
```

```
以前に試みた構成変更を繰り返します
zonecfg:kzone1> add anet;set lower-link=net1;set maxbw=2G;end
zonecfg:kzone1> commit
```

```
構成変更が成功したことがコマンド出力に表示されます
zone 'kzone1': Checking: Adding anet id=2
zone 'kzone1': Applying the changes
...
```


Oracle Solaris カーネルゾーンの移行

ゾーンの移行では、あるホストシステムの既存のゾーンが別のシステム上のゾーンに転送されます。

この章で扱う内容は、次のとおりです。

- 63 ページの「カーネルゾーンの移行について」
- 64 ページの「カーネルゾーン移行の実行に必要な権利」
- 66 ページの「使用する移行方式の決定」
- 67 ページの「カーネルゾーンの移行の要件」
- 69 ページの「カーネルゾーンの移行の準備」
- 76 ページの「コールド移行を使用したカーネルゾーンの移行」
- 78 ページの「ウォーム移行またはライブ移行用のサービスの有効化」
- 80 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」
- 83 ページの「ライブ移行を使用したカーネルゾーンの移行」

カーネルゾーンの移行について

あるシステムから別のシステムにゾーンを転送するには、`zoneadm migrate` コマンドを使用します。ユーザーの要件に応じてさまざまな状態のカーネルゾーンを移行できます。実行される移行のタイプはゾーンの状態によって決まります。

Oracle Solaris カーネルゾーン でサポートされる移行タイプは次のとおりです。

- **コールド移行** – ゾーン移行時に、ゾーンはソースホスト上で実行されていません。移行開始時にゾーンの状態が `installed` になっている必要があります。移行後、新しいホスト上での状態も同じになります。

大量のメモリーを使用するゾーンや、短い応答時間が必要なサービスを提供する、ライブ移行にはおそらく適さないゾーンを移行する場合は、この方式が有用です。

- **ウォーム移行** – ゾーンを移行する前に、ソースホスト上でゾーンを一時停止します。suspend リソースを構成する必要があります、移行開始時にゾーンの状態が

installed、補助状態が suspended になっている必要があります。移行後、新しいホスト上でもゾーンは suspended 補助状態になっているので、そこでゾーンを再開する必要があります。

この方式は、ゾーンブート時に長い起動時間が必要なアプリケーションを実行しており、ライブ移行は行えないようなゾーンで有用です。ウォーム移行ではゾーンの停止期間が必要になります。

- **ライブ移行** – ゾーンは移行時に、ソースホスト上でアクティブに実行されています。移行開始時にゾーンの状態が **running** になっている必要があります。移行時に移行対象ゾーンのメモリーの状態がターゲットホストにコピーされるので、新しいホスト上でゾーンは中断した時点から処理を再開できます。

停止時間を最小限に抑え、アプリケーションを **running** 状態のままにする必要がある状況では、この方式が有用です。移行プロセスでは、高い作業負荷に悪影響を与える可能性があるパフォーマンスの影響が発生することがあります。パフォーマンスへの影響が許容できない場合は、計画停止期間中にウォーム移行またはコールド移行を使用してください。

移行は、管理者または適切な承認を持つユーザーによって行われる必要があります。64 ページの「[カーネルゾーン移行の実行に必要な権利](#)」で説明しているように、root 以外の特定のユーザーが移行を実行できるようにすることが可能です。

また Oracle Solaris カーネルゾーンでは、退避と呼ばれる処理でホストシステムからすべてのカーネルゾーンを移行することもサポートされています。詳細は、[第6章「ターゲットホストへの Oracle Solaris カーネルゾーンの退避」](#)を参照してください。

カーネルゾーン移行の実行に必要な権利

root 以外のユーザーにゾーン権利プロファイルのサブセットを割り当てると、そのユーザーはカーネルゾーンの移行を実行できるようになります。大域ゾーンで、あるユーザーに次のプロファイルの1つ以上を割り当てると、そのユーザーはシステム上のすべてのゾーンを移行できるようになります。

- **Zone Migration** – ゾーン管理者が、インストール済みまたは実行中のゾーンの移行を実行できるようにします。
- **Zone Cold Migration** – ゾーン管理者が、インストール済みまたは一時停止済みのゾーンの移行を実行できるようにします。
- **Zone Configuration** – ゾーン管理者が、移行するゾーンに関してターゲットシステムを構成できるようにします。

ゾーンの権利プロファイルについては、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」を参照してください。ゾーン移行用の権利を割り当てる方法については、次の例を参照してください。

例 31 ある特定のゾーンのすべての移行を実行する承認をユーザーに付与

```
global1$ pfbash zonecfg -z kzone1
zonecfg:kzone1> add admin
zonecfg:kzone1:admin> set user=jdoe
zonecfg:kzone1:admin> set auths=migrate
zonecfg:kzone1:admin> end
zonecfg:kzone1> commit
```

承認とプロファイルを確認します。

```
global1$ zonecfg -z kzone1 info admin
admin:
    user: jdoe
    auths: migrate
$ auths jdoe
solaris.admin.wusb.read,solaris.mail.mailq,solaris.network.autoconf.read,solaris.zone.
migrate/kzone1
$ profiles jdoe
jdoe:
Zone Migration
Basic Solaris User
All
```

例 32 ホストシステム上のすべてのゾーンを移行する承認をユーザーに付与

```
global1$ pfbash usermod -P +"Zone Migration" -A +solaris.zone.migrate jdoe
```

承認とプロファイルを確認します。

```
global1$ auths jdoe
solaris.admin.wusb.read,solaris.mail.mailq,solaris.network.autoconf.read,solaris.zone.
migrate
global1$ profiles jdoe
jdoe:
Zone Migration
Basic Solaris User
All
```

例 33 ホストシステム上のゾーンを構成する承認をユーザーに付与

この例では、ターゲットシステム上で、ゾーン構成の作成に必要なプロファイルと承認をユーザー `jdoe` に割り当てています。割り当てられたプロファイルにより、ユーザーは任意のゾーン構成を作成、変更、および削除できるようになります。

```
global2$ pfbash usermod -P +"Zone Configuration" -A +solaris.zone.config jdoe
```

承認とプロファイルを確認します。

```
global2$ auths jdoe
solaris.admin.wusb.read,solaris.mail.mailq,solaris.network.autoconf.read,solaris.zone.
config
global2$ profiles jdoe
jdoe:
Zone Configuration
Basic Solaris User
All
```

権利プロファイルの割り当て方法や管理方法の詳細については、『[Oracle Solaris 11.4 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。

使用する移行方式の決定

コールド移行	カーネルゾーンがまだ実行されていない場合は、ゾーンを別のホストに移動させる必要があるときにコールド移行を使用します。
ライブ移行	<p>カーネルゾーンが実行中でサービスを提供している場合、通常はライブ移行が、ゾーンを別のホストに移動させる際のお勧めの方式になります。停止のスケジュールは必要なく、ネットワークの接続は維持され、アプリケーションは実行を続けることができます。</p> <p>ただし、次のいずれかの条件下ではライブ移行は適しません。</p> <ul style="list-style-type: none">■ ホスト間のネットワーク接続が低速であるか、帯域幅が貧弱である。■ ゾーン内で構成されたメモリーの量が非常に多い。■ 数百ミリ秒未満の非常に低い待機時間の応答時間がアプリケーションで必要となる。アプリケーションをテストしてこれを決定するようにしてください。■ デバイスが多重ホストディスク制御操作に有効化されており、ゾーン構成に <code>device</code> のプロパティ設定 <code>allow-mhd=true</code> が含まれている。
ウォーム移行	<p>コールド移行やライブ移行を実行できない場合、次の条件下ではおそらくウォーム移行がベターな選択肢になります。</p> <ul style="list-style-type: none">■ ゾーンが、共有ストレージ上で使用可能な十分なサイズの一時停止用 LUN を持っている。 一時停止イメージ用ストレージのサイズは、少なくとも <code>capped-memory:physical</code> プロパティ経由でカーネルゾーンに割り当てられたメモリーの量と同じ大きさにすべきです。■ 保守期間中に移行が発生する。 最大数分の停止時間が発生する可能性があります。■ 移行後の回復時間の短縮が必要とされる。 ウォーム移行では、ネットワーク帯域幅はさほど使用されません。ゾーンのネットワーク接続は途切れる可能性があります。アプリケーションのコールドリスタートは不要です。

ライブ移行、ウォーム移行のいずれも使用できない場合は、ゾーンを完全にシャットダウンしてコールド移行を使用してください。

カーネルゾーンの移行の要件

ライブ移行、ウォーム移行、コールド移行の要件はそれぞれ異なります。

カーネルゾーンのすべての移行方式での要件

カーネルゾーン用のコールド、ウォーム、およびライブ移行方式はすべて、次の要件を持っています。

- ソースとターゲットのホストシステムは同じアーキテクチャーファミリーである必要があります。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。
- ソースとターゲットのホストシステムはどちらも、[14 ページの「Oracle Solaris カーネルゾーンのソフトウェアおよびハードウェア要件」](#)で説明されている一般的なカーネルゾーン要件を満たす必要があります。
- ソースホストとターゲットホストの両方から、NFS、iSCSI、または LU のストレージ URI で定義された共有ストレージリソース経由でゾーンストレージにアクセスできる必要があります。

NFS ストレージ URI の詳細については、[108 ページの「NFS ストレージ URI とカーネルゾーン」](#)を参照してください。

ほかのタイプのストレージの共有ストレージ URI の詳細については、[suri\(7\) のマニュアルページ](#)および『[Creating and Using Oracle Solaris Zones](#)』の第 14 章、[「Oracle Solaris Zones on Shared Storage」](#)を参照してください。

- 移行に ssh を使用する予定の場合は、SSH でプロンプトが不要となるようにソースホストとターゲットホスト間の SSH 公開鍵認証を構成する必要があります。公開鍵のほかに、SSH プロンプトに 1 回以上リモートログインしたあとで公開鍵を記憶するように ssh-agent を構成する必要があります。『[Oracle Solaris 12 での Secure Shell アクセスの管理](#)』の「[How to Generate a Public/Private Key Pair for Use With Secure Shell](#)」を参照してください。
- ゾーン構成が、ソースホストとターゲットホストの両方で互換性があり、かつ一貫している必要があります。[71 ページの「移行および互換性のある構成について」](#)を参照してください。
- 実行中のカーネルゾーンのウォーム移行またはライブ移行を実行する場合は、[68 ページの「カーネルゾーンのウォーム移行およびライブ移行での追加要件」](#)で説明されている追加のシステム要件をソースホストとターゲットホストが満たす必要があります。

注記 - 共有ストレージを使用できない場合、`zoneadm migrate` コマンドを使用してゾーンを移行することはできません。ただし、統合アーカイブを作成し、新しいホストシステムでアーカイブからゾーンを作成し直すことはできます。『[Oracle Solaris 12 でのシステム復旧とクローン](#)』を参照してください。

カーネルゾーンのウォーム移行およびライブ移行での追加要件

注記 - 69 ページの「[カーネルゾーンのライブ移行での追加要件](#)」も参照してください。

ウォーム移行またはライブ移行を使用するには、67 ページの「[カーネルゾーンの移行の要件](#)」に記載された要件のほかに、ソースシステムとターゲットシステムの両方が次の要件を満たす必要があります。

- 両方のホストのオペレーティングシステムが Oracle Solaris 11.3 以上である必要があります。
- 同じプラットフォームの異なる CPU タイプ間でカーネルゾーンを移行する場合、72 ページの「[カーネルゾーンを異なる CPU または OS バージョンを持つシステムに移行するための準備](#)」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。
- SPARC ベースシステムのカーネルゾーンを Oracle Solaris 11.4 から Oracle Solaris 11.3 に移行する際に `host-compatible` プロパティを設定した場合、ソースホストとターゲットホストの値に互換性がある必要があります。72 ページの「[カーネルゾーンを異なる CPU または OS バージョンを持つシステムに移行するための準備](#)」を参照してください。
- カーネルゾーンが SR-IOV を使用している (`iovt` プロパティが有効になっている) 場合、カーネルゾーンを一時停止および再開し、ウォーム移行またはライブ移行を使用して移行を行う機能が、Oracle Solaris 11.4 以上を実行しているホストとゾーンに限定されます。詳細は、34 ページの「[SR-IOV が有効になった anet リソースを含むカーネルゾーンの移行について](#)」を参照してください。
- ウォーム移行およびライブ移行のソースおよびターゲットホストシステムでは、次のサービスのインスタンスが実行されている必要があります。
 - リモート管理デーモン (RAD) サービス、`svc:/system/rad:local` および `svc:/system/rad:remote`
 - NTP (Network Time Protocol) クライアント

これらのサービスの詳細については、78 ページの「[ウォーム移行またはライブ移行用のサービスの有効化](#)」を参照してください。

カーネルゾーンのライブ移行での追加要件

ライブ移行を使用するには、[67 ページの「カーネルゾーンの移行の要件」](#)と [68 ページの「カーネルゾーンのウォーム移行およびライブ移行での追加要件」](#)に記載された要件のほかに、ソースシステムとターゲットシステムの両方が次の要件を満たす必要があります。

- SPARC ベースシステム間のライブ移行では、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 M8 システム。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。
 - 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。
 - Fujitsu M10/SPARC M10 サーバー。Fujitsu M10 システムのプロダクトノートのファームウェア要件 (使用している構成に適したもの) に従ってください。
 - Fujitsu SPARC M12 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。
- 10 Gb 以上の Ethernet リンクが推奨されます。

停止時間の最小化が重要な場合は、ほかの移行も含め、そのリンク上でほかのトラフィックを制限することを検討してください。詳細は、[Oracle Solaris カーネルゾーンのベストプラクティス](#)に関する記事を参照してください。

カーネルゾーンの移行の準備

カーネルゾーンを移行する前に、`zoneadm migrate` コマンドを確認し、ソースゾーンとターゲットゾーンの互換性を維持する方法を学び、CPU や OS バージョンの違いの影響を検討し、暗号化デフォルトを使用して移行処理中のセキュリティーを確保する計画を立てます。

zoneadm migrate コマンドについて

各移行方式で使用される zoneadm コマンドは互いに似ています。移行の処理は、移行するゾーンの状態によって決まります。このコマンドの書式は次のとおりです。

```
zoneadm -z kzone migrate [-nq] -c cipher] [-t {auto|live}][ [ssh|rads|radg://user@host:port]
```

ここで:

-c cipher

移行用のセキュアな暗号化方式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。[75 ページの「セキュアな移行」](#)を参照してください。

-n

移行の、実行されない予行演習を行います。予行演習はすべてのタイプの移行で、両方のシステムから共有ストレージリソースにアクセスできることをチェックします。

- ライブ移行の場合、予行演習は完全な互換性の有無をチェックするので、チェックに合格した場合、ゾーンはリモートシステム上で正常に再開されます。
- ウォーム移行の場合、CPU やメモリーなどの設定に互換性がなければいけないのですが、予行演習はそれらの設定をチェックしません。移行時にゾーンが実行されていないからです。移行が完了したら、再開の前に必要に応じて設定を調整できます。
- コールド移行やウォーム移行の場合、予行演習は CPU の互換性をチェックしたり、ソースシステム上のゾーン構成とターゲットシステム上の任意の既存ゾーン構成を比較したりしません。コールド移行の場合、CPU やメモリーの互換性などをチェックする必要はありません。ゾーンは白紙の状態からブートされるからです。移行が完了したら、ゾーン構成の設定を必要に応じて調整できます。

-q

非出力モード。移行処理中にステータスが報告されないことを指定します。

-t {live | auto}

移行のタイプを指定します。-t のデフォルト値である auto を指定した場合、zoneadm migrate コマンドは、適切な移行タイプをゾーンの状態に基づいて自動的に決定できるようになります。-t live オプションを使用すると、実行中のゾーンのみに移行を制限できます。実行中でないゾーンで -t live を指定した場合、そのゾーンの移行は行われません。

```
ssh|rads|radg://user@host:port
```

ゾーンをターゲットシステム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。radg スキームは Generic Security Services API (GSS-API) を使用します。RAD クライアントとターゲットホストが Kerberos 向けに構成されている場合は、radg を指定します。

ホスト名のみを指定した場合、スキームのデフォルトは rads になり、user のデフォルトは現在のユーザーになり、port のデフォルトは標準 RAD ポート 12302 になります。

詳細は、『[Remote Administration Daemon Client User's Guide](#)』の「[Connecting in Python to a RAD Instance by Using a URI](#)」を参照してください。

migrate コマンドの詳細については、[zoneadm\(8\)](#) のマニュアルページを参照してください。

移行および互換性のある構成について

カーネルゾーンの構成は、ゾーンを切り離してから接続した場合と同様に、移行ターゲットホストの環境と互換性を持っている必要があります。ウォーム移行やライブ移行後に新しいホスト上でブートされるゾーンは、保存されたメモリー状態から再開されるので、特定の設定を想定しています。何らかの非互換性があると、移行が失敗します。

カーネルゾーンを同一の別のシステムに移行する場合に、両方のホストからアクセス可能なストレージ URI がすべてのストレージ参照で使用されていれば、移行される構成は変更しなくても互換性があるはずです。

ゾーンのストレージがローカルの場合、zoneadm migrate コマンドを使用できません。ローカルストレージデバイスがブートに必要な場合にそれらをゾーン構成から削除するか、または『[Creating and Using Oracle Solaris Zones](#)』の「[How to Move a Zone To a Shared Storage Configuration](#)」の説明に従ってストレージを共有に変換したあとで、zoneadm migrate を使用できます。

別の方法として、統合アーカイブを使用してゾーンを移動することもできます。詳細は、『[Oracle Solaris 12 でのシステム復旧とクローン](#)』を参照してください。

次のリソースやプロパティは、ソースホストとターゲットホストのゾーン構成で同じになっている必要があります。

- capped-memory:physical の値として指定されたメモリー量
- capped-memory:pagesize-policy プロパティの値。
- virtual-cpu:ncpus に指定された仮想 CPU の数
- device リソースで指定されたディスクデバイスの共有ストレージ URI と id
- net または anet リソースに指定された仮想 NIC のプロパティ

移行の前にターゲットホスト上でゾーンを構成した場合、そのターゲットホスト版のゾーン構成がゾーンのブートに使用されます。その構成が現在のゾーン構成と互換性がない場合は、エラーが返されます。また、ゾーンの暗号化鍵も一致する必要があります。105 ページの「暗号化鍵とホストデータ」を参照してください。

移行の前にターゲットホスト上でゾーンを構成しない場合は、ソースホストからゾーン構成がエクスポートされ、それがターゲットホストにインポートされます。移行を実行するユーザーは、Zone Configuration 権利プロファイルと `solaris.zone.configuration` 承認を持っていないと、ターゲットホスト上でゾーン構成を作成できません。詳細は、64 ページの「カーネルゾーン移行の実行に必要な権利」を参照してください。

ターゲットホスト環境が同一でない場合は、次のガイドラインに従ってください。

- システムの CPU が異なっている場合にウォーム移行またはライブ移行を実行するには、`cpu-arch` を移行クラスに設定する必要があります。72 ページの「カーネルゾーンを異なる CPU または OS バージョンを持つシステムに移行するための準備」を参照してください。コールド移行では、このプロパティを設定する必要はありません。
- SPARC ベースシステムのソースホストとターゲットホストで異なるバージョンの Oracle Solaris が実行されている場合、両方のホストでサポート可能な Oracle Solaris 機能を指定するために `host-compatible` プロパティを設定する必要があります。72 ページの「カーネルゾーンを異なる CPU または OS バージョンを持つシステムに移行するための準備」を参照してください。コールド移行では、このプロパティを設定する必要はありません。
- ソースホストで Oracle Solaris 11.4 が実行され、ターゲットホストで Oracle Solaris 11.3 が実行されている場合は、`pagesize-policy` プロパティをクリアする必要があります。26 ページの「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」を参照してください。

カーネルゾーンを異なる CPU または OS バージョンを持つシステムに移行するための準備

カーネルゾーンは、CPU は異なるがプラットフォームは同じであるような、ほかのホストシステムに移行できます。たとえば、カーネルゾーンを SPARC T4 サーバーから SPARC T7 サーバーに移行したり、Intel Nehalem ベースのサーバーから Haswell プロセッサに基づくサーバーに移行したりできます。これはクロス CPU 移行と呼ばれます。ゾーンを SPARC ベースの環境から x86 ベースの環境に移行することはできません。

カーネルゾーンを、ソースシステムとは異なるプロセッサを搭載したターゲットシステムに移行する場合は、ウォーム移行やライブ移行用にカーネルゾーンを一時停止する前に、ゾーンの準備を整えてリポートする必要があります。

SPARC ベースおよび Intel ベースのシステムの場合、`cpu-arch` リソースタイプを使用してプロセッサ機能の共通セットを定義する移行クラスを指定すれば、ソースシステムとターゲットシステムでゾーンを正しく動作させることができます。

AMD システムでは `cpu-arch` リソースタイプを使用できません。

SPARC ベースシステムでのみ、さらに `host-compatible` リソースタイプを使用して互換性レベルを設定することで、特定のプロセッサで有効化される Oracle Solaris 機能が、ソースホストとターゲットホストの大域ゾーン内と同じレベルでサポートされるようになります。`host-compatible` を設定しなかった場合、Oracle Solaris 11.2 の機能のみがゾーンに対して可視となります。Oracle Solaris 11.2 は、カーネルゾーンで実行可能なもっとも古いバージョンの Oracle Solaris ですが、SSM や DAX などの機能はサポートしていません。

注記 - ライブ移行が成功するためには、ソースホストとターゲットホストの両方で実行されている Oracle Solaris リリースが、ゾーン構成の `cpu-arch` と `host-compatible` の値をサポートしている必要があります。ある機能が異なる方法で有効化されていた場合、その機能が両方のリリースでサポートされていたとしても、ホスト間でゾーンをライブ移行できません。

たとえば、ソースホストのゾーン構成で `host-compatible=native` が設定され、ターゲットホストのゾーン構成で `host-compatible=level1` が設定されていた場合、両方のホストで Oracle Solaris 11.4 が実行されていたとしても、ライブ移行は失敗します。

異なる SPARC アーキテクチャー間での移行のための、`cpu-arch` および `host-compatible` リソースタイププロパティの詳細については、『[Oracle Solaris Zones Configuration Resources](#)』の「[Kernel Zone Migration Class and Host Compatibility Level \(solaris-kz Only\)](#)」を参照してください。

異なる Intel アーキテクチャー間での移行で `cpu-arch` リソースタイプを設定する方法の詳細については、『[Oracle Solaris Zones Configuration Resources](#)』の「[Cross-CPU Migration Classes \(solaris-kz\)](#)」を参照してください。

カーネルゾーンの `cpu-arch` プロパティが移行クラスに設定されていない場合、カーネルゾーンの CPU アーキテクチャーはホストシステムと同じになります。

注記 - カーネルゾーンホストは常に、互換性のない CPU タイプで以前に一時停止されたゾーン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、カーネルゾーンはブートしません。

例 34 SPARC ベースシステム上のカーネルゾーンの `cpu-arch` および `host-compatible` リソースの確認および設定

次の例は、カーネルゾーン `kzone1` の `cpu-arch` および `host-compatible` リソースタイプ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 host-compatible
cpu-arch: generic
host-compatible: native
zonecfg:kzone1> set cpu-arch=migration-class1
zonecfg:kzone1> set host-compatible=adi
zonecfg:kzone1> info cpu-arch host-compatible
cpu-arch: migration-class1
host-compatible: adi
zonecfg:kzone1> exit
```

例 35 互換性のない SPARC CPU アーキテクチャーのために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 プロパティは実際の CPU アーキテクチャーを示すデフォルト値を使用しています。cpu-arch プロパティの値はホスト間で一貫性がないので、『[Oracle Solaris Zones Configuration Resources](#)』の「[Kernel Zone Migration Class and Host Compatibility Level \(solaris-kz Only\)](#)」で説明されている移行クラス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: Cannot resume guest on target host.
error: Guest's migration class is SPARC-T4, host's is SPARC-T5. Please check cpu-arch
setting in zone config or in host LDom config.
2016-08-18 18:27:53 error: request failed: failed to create VM: Operation not supported
```

例 36 Intel システム上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 リソースタイプ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-class4
zonecfg:kzone1> info cpu-arch
cpu-arch: migration-class4
zonecfg:kzone1> exit
global$ zoneadm kzone1 reboot
```

例 37 互換性のない Intel CPU アーキテクチャーのためにライブ移行が失敗する

この例は、Haswell ベースのホスト global1 と Sandy Bridge ベースのホスト global2 の間でのライブ移行の試みを示しています。これは、Oracle Server X5-2 などから CPU として Sandy Bridge プロセッサを搭載した別のタイプのサーバーへの移行時に、cpu-arch を設定していなかった場合に発生します。cpu-arch プロパティを、『[Oracle Solaris Zones Configuration Resources](#)』の「[Cross-CPU Migration Classes \(solaris-kz\)](#)」で説明されている移行クラス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 live migration compatibility.
zoneadm: zone 'kzone1': configuration check failed:
See /var/log/zones/kzone1.messages for debug output
error: cannot resume as current CPU migration class value: sandybridge does not match the
suspended value haswell
2016-08-15 01:05:55 error: request failed: failed to create VM: Invalid argument

```

セキュアな移行

デフォルトでは、移行メモリー転送データはソースホストとターゲットホスト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* を使用すると、特定の暗号化方式を指定したり、暗号化を無効にしたりできます。

zoneadm migrate の -c *cipher* オプションで使用できる値は、次のとおりです。

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。

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。

none

暗号化を無効にします。

暗号化方式を指定しなかった場合、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 1 つの方式が自動的に選択されます。

例 38 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。

```

global1$ zoneadm -z kzone1 migrate 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 39 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` のライブ移行を実行する試みを示しています。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 40 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: aes-128-cbc
# echo $?
0
```

ヒント - 移行されたカーネルゾーンのブートに必要な暗号化鍵が失われないようにするには、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$ pfbash zonecfg -z kzone1 export -f /net/example/path/kzone1.cfg
global$ zonecfg -z kzone1 -f /net/example/path/kzone1.cfg
```

ゾーンのブートを可能にする暗号化鍵については、[105 ページの「暗号化鍵とホストデータ」](#)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

コールド移行では、実行中でないゾーンが切り離され、別のホスト上で接続されます。そのホスト上でゾーンをリブートできます。

コールド移行の詳細については、[63 ページの「カーネルゾーンの移行について」](#)および [66 ページの「使用する移行方式の決定」](#)を参照してください。

▼ コールド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 [67 ページの「カーネルゾーンの移行の要件」](#) で説明した要件をソースホストとターゲットホストが満たしていることを確認します。

1. カーネルゾーンを移行する権利が割り当てられた管理者になります。
詳細は、[64 ページの「カーネルゾーン移行の実行に必要な権利」](#) を参照してください。

2. 移行するゾーンが実行中の場合は、そのゾーンをシャットダウンします。

```
source-host$ zoneadm -z kzone shutdown
```

3. (オプション) 状態が **installed** になっていることを確認します。

次に出力例を示します。

```
source-host$ zoneadm -z kzone list -v
ID NAME          STATUS    PATH    BRAND    IP
- kzone          installed -       solaris-kz  excl
```

4. (オプション) `ssh://URI` を使用してターゲットホストに接続する予定である場合は、プロンプトなしの SSH 認証をテストします。

ターゲットホストで、`date` などのコマンドを `ssh` 経由で実行します。

```
source-host$ ssh target-host date
Tue Oct  4 17:07:55 MDT 2016
```

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 12 での Secure Shell アクセスの管理』](#) の「[How to Generate a Public/Private Key Pair for Use With Secure Shell](#)」を参照してください。

5. (オプション) 移行の予行演習を実行し、条件が適切に設定されていることを確認します。

次に出力例を示します。

```
source-host$ zoneadm -z kzone migrate -n ssh://user@target-host
zoneadm: zone 'kzone': Importing zone configuration.
zoneadm: zone 'kzone': Attaching zone.
zoneadm: zone 'kzone': Dry-run migration successful.
zoneadm: zone 'kzone': Cleaning up.
```

6. 移行を実行します。

次に出力例を示します。

```
source-host$ zoneadm -z kzone migrate ssh://user@target-host
zoneadm: zone 'kzone': Importing zone configuration.
```

```
zoneadm: zone 'kzone': Attaching zone.
zoneadm: zone 'kzone': Migration successful.
```

7. (オプション) ターゲットホストでゾーンをブートします。

```
target-host$ zoneadm -z kzone boot
```

例 41 ローカルストレージに起因するコールド移行の予行演習の失敗

次の例は、カーネルゾーンが実行中でないことの確認と、コールド移行の予行演習を示しています。カーネルゾーン z3kz でローカルストレージが使用されているので、この予行演習は失敗します。

```
root@global3 $ zoneadm list -cv
ID NAME          STATUS  PATH          BRAND  IP
0  global         running /             solaris shared
-  z3kz          installed -             solaris-kz excl

root@global3 $ zoneadm -z z3kz migrate -n ssh://global5
zoneadm: zone 'z3kz': configuration check failed: The storage property
dev:/dev/zvol/dsk/rpool/VARSHARE/zones/z3kz/disk0 is not a shared storage URI.
```

例 42 構成変更後の正常なコールド移行

次の例は、ゾーン z3kz のゾーン構成からのブート用でないローカルストレージデバイスの削除と、正常な移行を示しています。使用されるゾーン構成は、移行するゾーンからのものであり、ターゲットホストでインポートされます。

```
root@global3 $ zonecfg -z z3kz 'remove device id=1;commit;exit'
root@global3 $ zoneadm -z z3kz migrate ssh://global5
zoneadm: zone 'z3kz': Importing zone configuration.
zoneadm: zone 'z3kz': Attaching zone.
zoneadm: zone 'z3kz': Migration successful.
```

参照 以前のバージョンの Oracle Solaris に移行する場合は、[例9「以前の Oracle Solaris リリースにカーネルゾーンを移行する前に pagesize-policy プロパティをクリアする」](#)を参照してください。

ウォーム移行またはライブ移行用のサービスの有効化

67 ページの「[カーネルゾーンの移行の要件](#)」で説明したように、ウォーム移行またはライブ移行を実行する際には、ソースホストとターゲットホストでいくつかのサービスが実行されている必要があります。このセクションでは、必要なサービスを確認し、それらを必要に応じて有効化する方法について説明します。

Oracle Solaris サービスの管理の追加情報については、『[Oracle Solaris 11.4 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。

▼ カーネルゾーンの移行に必要なサービスを確認および有効化する方法

1. ゾーン管理者になります。

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」を参照してください。

2. `ssh:// URI` を使用してターゲットホストに接続する予定である場合は、プロンプトなしの SSH 認証をテストします。

ターゲットホストで、`date` などのコマンドを `ssh` 経由で実行します。

```
source-host$ ssh target-host date
Tue Oct  4 17:07:55 MDT 2016
```

パスワードの入力を求められる場合は、SSH ログインの構成が完了していません。

『[Oracle Solaris 12 での Secure Shell アクセスの管理](#)』の「[How to Generate a Public/Private Key Pair for Use With Secure Shell](#)」を参照してください。

3. NTP クライアントサービスが実行されていることを確認します。

```
source-host$ svcs ntp
STATE          STIME      FMRI
online         Jun_27    svc:/network/ntp:default
```

```
source-host$ ssh target-host svcs ntp
STATE          STIME      FMRI
online         Aug_09    svc:/network/ntp:default
```

NTP サービスが実行されていない場合は、NTP サービスを起動します。追加情報については、『[Oracle Solaris 12 でのクロック同期の管理](#)』の「[Oracle Solaris システム上で NTP を設定する方法](#)」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 RAD サービスが実行されていることを確認します。

- `ssh:// URI` を使用する予定の場合は、`rad:local` サービスが実行されている必要があります。
- `rads://` または `radg://` を使用する予定の場合は、`rad:remote` サービスが実行されている必要があります。

- a. 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表示します。

次の出力例では、ソースホストとターゲットホストの両方で `rad:remote` サービスが無効になっています。

ソースホストで RAD サービスを確認します。

```
source-host$ svcs rad
STATE          STIME      FMRI
disabled      10:09:15  svc:/system/rad:remote
online        10:09:18  svc:/system/rad:local
```

```
online          10:09:18 svc:/system/rad:local-http
```

ターゲットホストで RAD サービスを確認します。

```
source-host$ ssh target-host svcs rad
STATE          STIME          FMRI
disabled       Jun_23         svc:/system/rad:remote
online         Jun_23         svc:/system/rad:local
online         Jun_23         svc:/system/rad:local-http
```

- b. 無効になっている RAD サービスが存在する場合は、そのサービスを有効にします。

次のコマンド例では、ソースホストとターゲットホストの両方で rad:remote サービスを有効化しています。

ソースホストで RAD サービスを有効化します。

```
source-host$ svcadm enable rad:remote
```

ターゲットホストで RAD サービスを有効化します。

```
source-host$ ssh target-host svcadm enable rad:remote
```

- c. **ステップ 4a** を繰り返し、すべての RAD サービスが有効になっていることを確認します。

5. (オプション) ライブ移行を実行する予定の場合は、ソースホストとターゲットホストでカーネルゾーン移行サービスを起動します。

```
source$ svcadm enable -rs svc:/network/kz-migr:stream
source$ ssh target svcadm enable -rs svc:/network/kz-migr:stream
```

ウォーム移行を使用したカーネルゾーンの移行

zoneadm suspend コマンドのあとで zoneadm migrate コマンドを使用すると、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、ゾーン構成が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必要があります。移行の前にターゲットホストでゾーン構成を作成する場合は、その構成が使用されます。それ以外の場合は、移行するゾーンの構成が使用されます。

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を持つように構成されている必要があります。solaris-kz(7) のマニュアルページおよび『[Creating and Using Oracle Solaris Zones](#)』の第 14 章、「[Oracle Solaris Zones on Shared Storage](#)」を参照してください。

移行時には、ターゲットシステムからゾーンの共有ストレージにアクセスできることの確認、ソースシステムでのカーネルゾーンの切り離し、および宛先システムでのゾーンの接続、が自動的行われます。

注記 - 同一のプロセッサを使用していないシステムを移行する場合は、一時停止の前に、72 ページの「[カーネルゾーンを異なる CPU または OS バージョンを持つシステムに移行するための準備](#)」の説明に従って `cpu-arch` プロパティを設定します。

移行が完了したら、ターゲットシステム上でゾーンを手動で再開できます。再開については、83 ページの「[ウォーム移行後のカーネルゾーンの再開について](#)」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
- 67 ページの「[カーネルゾーンの移行の要件](#)」で説明した要件をソースホストとターゲットホストが満たしていることを確認します。
 - 78 ページの「[ウォーム移行またはライブ移行用のサービスの有効化](#)」の説明に従って、必要なサービスが使用可能になっていることを確認します。
 - ソースホストとターゲットホストが同一でない場合は、71 ページの「[移行および互換性のある構成について](#)」を参照してください。
1. **カーネルゾーンを移行する権利が割り当てられた管理者になります。**
詳細は、64 ページの「[カーネルゾーン移行の実行に必要な権利](#)」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。**
次に出力例を示します。

```
source-host$ zonecfg -z kzone info suspend
suspend:
  storage: iscsi://system/luname.naa.501337600144f0dbf8af1900
```

ターゲットホストは、同じ URI を使用してこの場所にアクセスできる必要があります。`suspend` リソースが構成されていない場合は、36 ページの「[suspend リソースタイプの構成](#)」を参照してください。
 3. **ソースホストの大域ゾーンで、移行するカーネルゾーンを一時停止します。**

```
source-host$ zoneadm -z kzone suspend
```

注記 - 一時停止の処理では、メモリーを含むゾーンの状態がディスクに書き込まれるので、完了までに長い時間がかかる可能性があります。

4. (オプション) 移行の予行演習を実行し、共有ストレージにアクセスできることを確認します。

次に出力例を示します。

```
source-host$ zoneadm -z kzone migrate -n ssh://user@target-host
zoneadm: zone 'kzone': Importing zone configuration.
zoneadm: zone 'kzone': Attaching zone.
zoneadm: zone 'kzone': Dry-run migration successful.
zoneadm: zone 'kzone': Cleaning up.
```

予行演習で、共有ストレージへのアクセスに関する問題が発生した場合は、それらの問題を修正してから次に進みます。『[Creating and Using Oracle Solaris Zones](#)』の第14章、「[Oracle Solaris Zones on Shared Storage](#)」を参照してください。

5. ゾーンをターゲットホストに移行します。

このステップでは、ターゲットホストシステムでソースホストと同じゾーン構成を使用してゾーンが構成され、ターゲットホストでゾーンが接続されます。

```
source-host$ zoneadm -z kzone migrate rad-uri:user@target-host
```

- ssh を使用して移行する場合:

```
source-host$ zoneadm -z kzone migrate ssh://root@target-host
zoneadm: zone 'kzone': Importing zone configuration.
zoneadm: zone 'kzone': Attaching zone.
zoneadm: zone 'kzone': Migration successful.
```

- rads を使用して移行する場合:

```
source-host$ zoneadm -z kzone migrate rads://root@target-host:12302
```

6. (オプション) 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。

```
target-host$ zoneadm -z kzone boot
```

例 43 ウォーム移行での構成チェックの失敗

この例は、ローカルパス上の suspend リソースによる移行の試みを示しています。

```
global3$ zoneadm -z z1kz migrate ssh://global5
zoneadm: zone 'z1kz': configuration check failed: suspend path resource must be an NFS path
```

例 44 カーネルゾーンの一時停止とウォーム移行

この例は、suspend リソースを表示し、ゾーンを一時停止し、カーネルゾーンの補助状態 (z2kz では suspended) を含めてゾーンの情報を一覧表示するためのコマンドと、一時停止されたカーネルゾーンの正常な移行を示しています。ターゲットホスト上の移行先のゾーンで uptime コマンドを発行すると、ゾーンの実行時間が表示されますが、これには、ソースホスト上でのブートされてからの時間も含まれています。

```
global3$ zonecfg -z z2kz info suspend
suspend:
  storage: iscsi://system/luname.naa.501337600144f0dbf8af1900
global3$ zoneadm -z z2kz suspend

global3$ zoneadm list -cp
0:global:running:/::solaris:shared:-:none:
-:z2kz:installed:/system/volatile/zones/z2kz/zonepath:890d94b7-23c7-48c8-922e-ede10c3d1ac6:
solaris-kz:excl:-:solaris-kz:suspended

global3$ zoneadm -z z2kz migrate ssh://global5
zoneadm: zone 'z2kz': Importing zone configuration.
zoneadm: zone 'z2kz': Attaching zone.
zoneadm: zone 'z2kz': Migration successful.

global3$ ssh global5 zlogin z2kz uptime
12:02pm up 2 day(s),  2:55,  0 users,  load average: 0.04, 0.04, 0.03
```

ウォーム移行後のカーネルゾーンの再開について

ゾーンが一時停止または停止されたあとでソースシステムのゾーン構成が変更された場合、移行したゾーンのブートが失敗する可能性があります。また、実行中のゾーンに対してライブゾーン構成変更が行われたあとでゾーンが一時停止された可能性もありますが、その場合、永続的なゾーン構成にはその構成変更は反映されていません。このため、ソースとターゲットのゾーン構成間やシステム間での互換性チェックは一切行われません。ストレージのみがチェックされます。

構成の互換性が保たれていることを移行前に確認していた場合、ターゲットシステム上の移行したカーネルゾーンを再開できるはずです。ソースシステムとターゲットシステム間の互換性のため、つまりゾーン構成の違いのために、ターゲットシステム上の一時停止されたゾーンを再開できない場合は、ターゲットシステム上のゾーン構成を変更したあとで、もう一度再開を試みてください。boot コマンドで -R オプションを指定して非再開モードでゾーンをブートすることもできますが、その場合はゾーンの一時停止イメージに保存された情報が使用されないため、それはもうウォーム移行ではありません。

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の状態が移行先のゾーン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。ネットワーク接続は維持されます。

ライブ移行は、停止時間を最小限に抑える必要のあるアプリケーションや、アプリケーションがサービスを提供し続ける必要がある場合に使用できます。

▼ 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、67 ページの「カーネルゾーンの移行の要件」と 68 ページの「カーネルゾーンのウォーム移行およびライブ移行での追加要件」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。
 - 78 ページの「ウォーム移行またはライブ移行用のサービスの有効化」の説明に従って、必要なサービスが使用可能になっていることを確認します。
 - ソースホストとターゲットホストが同一でない場合は、71 ページの「移行および互換性のある構成について」を参照してください。

1. カーネルゾーンを移行する権利が割り当てられた管理者になります。

詳細は、64 ページの「カーネルゾーン移行の実行に必要な権利」を参照してください。

2. 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。

```
source-host$ zoneadm list -cv
ID NAME          STATUS    PATH    BRAND      IP
0  global         running  /       solaris    shared
1  kzone          running  -       solaris-kz excl
```

3. ソースホストで予行演習を起動します。

この処理では、ライブ移行の実行前にカーネルゾーンの構成がテストされます。次に出力例を示します。

```
source-host$ zoneadm -z kzone migrate -n rad-uri:user@target-host
zoneadm: zone 'kzone': Importing zone configuration.
zoneadm: zone 'kzone': Attaching zone.
zoneadm: zone 'kzone': Booting zone in 'migrating-in' mode.
zoneadm: zone 'kzone': Checking migration compatibility.
zoneadm: zone 'kzone': Cleaning up.
zoneadm: zone 'kzone': Dry-run migration successful.
```

4. カーネルゾーンを移行します。

```
source-host$ zoneadm -z kzone migrate rad-uri:user@target-host
zoneadm: zone 'kzone': Importing zone configuration.
zoneadm: zone 'kzone': Attaching zone.
zoneadm: zone 'kzone': Booting zone in 'migrating-in' mode.
...
zoneadm: zone 'kzone1': Halting and detaching zone on source host.
zoneadm: zone 'kzone': Migration successful.
```

5. ゾーンがターゲットホストに移行されたことを確認します。

```
target-host:$ zoneadm list -cv
ID NAME          STATUS    PATH    BRAND      IP
0  global         running  /       solaris    shared
1754 kzone          running  -       solaris-kz excl
...
```

例 45 ライブ移行を使用してカーネルゾーンを新しいホストに移行する

次の例は、ソースホスト `global` からターゲットホスト `global2` へのカーネルゾーン `kzone1` のライブ移行を示しています。構成はターゲットホスト上で事前に作成されました。

```
global$ zoneadm -z kzone1 migrate ssh://global2
zoneadm: zone 'kzone1': Using 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': Waiting for migration to complete.
zoneadm: zone 'kzone1': Migration successful.
zoneadm: zone 'kzone1': Halting and detaching zone.
```

例 46 ライブ移行の予行演習の失敗

この例は、ソースホスト `global1` とターゲットホスト `global2` の間で失敗した予行演習の移行を示しています。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(8) on the
zone 'kzone1': destination host.
zoneadm: zone kzone1: call to zoneadmd(8) failed: zoneadmd(8) returned an error 9 (zone
state change failed)
```

例 47 2つの異なる anet 構成を持つホスト間のライブ移行

次の例は、anet 構成が異なるホスト間でのライブ移行を示しています。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。

移行の前に、ターゲットホスト上でゾーン構成が作成され、anet リソースが変更されています。テストのために予行演習が実行されています。

```
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
```


ターゲットホストへの Oracle Solaris カーネルゾーンの退避

カーネルゾーン退避を使用すれば、`running`、`installed` のいずれかの状態にあるカーネルゾーンのすべてを、あるシステムから別のシステムに転送できます。退避の主な用途は、停止時間のないシステム保守です。この章では、カーネルゾーン退避の要件やタスクについて説明します。

カーネルゾーン退避について

カーネルゾーン退避は、実行中のすべてのカーネルゾーンをシステム外へ一度にライブ移行する処理であり、オプションで、それらのゾーンをあとでシステムに戻すこともできます。すべてのカーネルゾーンをホストシステムからほかのシステムに一時的にライブ移行すれば、ホストシステムで保守を実行する際に、それらのカーネルゾーン内で実行されているアプリケーションを停止する必要がなくなります。オプションで、実行中でないカーネルゾーンと `installed` 状態の `solaris` ゾーンを含む、すべてのゾーンを退避することもできます。

退避では、宛先へのゾーンの移行の調整や実行に Remote Administration Daemon (RAD) が使用されるので、ソースホストとターゲットホストで RAD サービスが実行されている必要があります。RAD の詳細については、[rad\(8\)](#) のマニュアルページを参照してください。

カーネルゾーン退避のステップ

カーネルゾーン退避の手順概要:

1. ソースホストとターゲットホストで要件が満たされていることを確認します。
[89 ページの「カーネルゾーン退避の要件」](#) を参照してください。
2. 移行する各カーネルゾーンの宛先ホストを設定するため、SMF サービスのプロパティを設定します。

90 ページの「[カーネルゾーン退避用のターゲットホストの設定](#)」を参照してください。

3. ソースホストを保守モードに切り替え、実行中でないゾーンについて接続、ブート、または移行の受け入れが発生しないようにします。

91 ページの「[カーネルゾーン退避の準備としての保守モードの設定](#)」を参照してください。

4. `evacuate` コマンドを実行し、実行中のカーネルゾーンを事前に設定された宛先に移行します。

92 ページの「[カーネルゾーンの退避](#)」を参照してください。

5. ソースホストでシステムの保守を実行し、リブートします。
6. ソースホストシステムで保守モードを終了します。

94 ページの「[カーネルゾーン退避後の保守モードの終了](#)」を参照してください。

7. 必要であれば、退避されたゾーンをソースホストに戻します。また退避は、新しいホストにゾーンを永続的に移行するためにも使用できます。

94 ページの「[退避されたカーネルゾーンを元のシステムに戻す](#)」を参照してください。

sysadm ユーティリティーと Oracle Solaris カーネルゾーン

`sysadm` ユーティリティーではすべての退避タスクを実行できます。

- ゾーンへの接続、ゾーンのブート、または移行ゾーンの受け入れを禁止する保守モードにシステムを切り替えることで、移行の準備を整えるには、`sysadm maintain` コマンドを使用します。
- 退避操作でゾーンの実際の移行を実行するには、`sysadm evacuate` コマンドを使用します。

完全な情報については、[sysadm\(8\)](#) のマニュアルページを参照してください。

sysadm maintain コマンド

`sysadm maintain` コマンドで使用できるオプションは、次のとおりです。

- e 保守モードを終了します。
- l 保守モードの現在のステータスを表示します。

- m "message-text" 保守の理由などを示すために指定できるメッセージ。
- s 保守モードを開始し、ソースホスト上で新しいゾーンが実行されないようにします。

保守状態はシステムをリブートしても持続し、すべてのブート環境に適用されるので、`sysadm maintain -e` コマンドを使用して明示的に保守モードを終了させる必要があります。

sysadm evacuate コマンド

`sysadm evacuate` コマンドで使用できるオプションは、次のとおりです。

- a すべてのゾーン、つまり実行中でないゾーンと実行中のゾーンの両方、が退避されます。
- n 退避がどのように実行されるかを示すための、退避の実行されない予行演習。
- q エラーのみを表示する非冗長出力。
- r 退避されたゾーンをソースホストに戻します。
- v 退避の進捗を表示する冗長出力。

カーネルゾーン退避の要件

カーネルゾーン退避を実行する場合は、次の要件に従ってください。

- **移行の要件** – ソースホストとターゲットホスト、およびそれらの構成済みストレージは、[67 ページの「カーネルゾーンの移行の要件」](#)で説明されている要件を満たす必要があります。
- **Oracle Solaris のバージョン** – 実行中のカーネルゾーン、および実行中でないインストール済みのカーネルゾーンと非大域ゾーンを退避したり戻したりする機能など、すべての機能のフルサポートを得るには、ソースホストとターゲットホストの両方で Oracle Solaris 11.4 が実行されている必要があります。installed 状態のゾーンはコールド移行を使用して退避されます。

Oracle Solaris 11.4 が実行されているシステムから Oracle Solaris 11.3 が実行されているシステムに実行中のカーネルゾーンを退避したり退避したゾーンを戻したりする機能もサポートされます。ただし、コールド移行は Oracle Solaris 11.3 ではサ

ポートされないので、実行中でないゾーンを `-a` オプションで退避することはできません。

- **ライブ移行の要件** – 退避処理ではライブ移行が使用されるので、ソースホストとターゲットホストはライブ移行の要件も満たす必要があります。詳細は、[68 ページの「カーネルゾーンのウォーム移行およびライブ移行での追加要件」](#)を参照してください。
- **サービスの有効化** – [78 ページの「ウォーム移行またはライブ移行用のサービスの有効化」](#)の説明に従ってサービスを有効化する必要があります。
- **ユーザーの権利と承認** – 退避を実行するユーザーは、[64 ページの「カーネルゾーン移行の実行に必要な権利」](#)で説明されている権利プロファイルと承認を持っている必要があります。さらにユーザーは、`sysadm maintain` コマンドを実行するための Maintenance and Repair 権利プロファイルも持っている必要があります。
- **非対話型認証** – ソースホストと各宛先の間で、RAD トランスポート経由での双方向の非対話型認証を構成する必要があります。つまり、退避を実行するユーザーは、パスワードの入力や鍵のフィンガープリントの確認を要求されることなしに、ソースホストからターゲットホストおよびターゲットホストからソースホストに RAD URI を使用して接続できる必要があります。単純なホスト名、ドメイン名を含む完全修飾ホスト名の両方を使用して非対話型の接続を行えることを確認しなければいけない可能性があります。

ソースホストとターゲットホスト間での SSH 公開鍵認証は、非対話型認証を実行するための方法の 1 つです。詳細は、『[Oracle Solaris 12 での Secure Shell アクセスの管理](#)』の「[How to Generate a Public/Private Key Pair for Use With Secure Shell](#)」を参照してください。

カーネルゾーン退避用のターゲットホストの設定

ホスト間の非対話型認証の設定が完了したら、ゾーン委任リスタートサービスの `evacuation/target` SMF プロパティを設定することで、ターゲットホストの RAD サービスに接続するための URI を指定します。

退避されるゾーンの受信と実行に適したターゲットホストを選択する必要があります。移行するゾーンのいずれかが、構成を変更しないとターゲットホスト上で実行できない場合は、退避の前にターゲットホストでゾーン構成を作成するようにしてください。[71 ページの「移行および互換性のある構成について」](#)を参照してください。

`evacuation/target` プロパティは、ソースホストのすべてのゾーンに適用されるように設定することも、個々のゾーンごとに異なるターゲットホストに設定することもできます。また、すべてのゾーン用にプロパティを設定したあと、個々のゾーンサービスインスタンスで設定をオーバーライドすることもできます。

すべてのゾーンに対して `evacuation/target` プロパティを設定した場合、ゾーン委任リスタートサービス `svc:/system/zones/zone` をリフレッシュする必要があります。

ます。特定のゾーンに対してターゲットを設定する場合、`evacuation/target` プロパティを設定したあと、そのゾーン委任リスタータサービス `svc:/system/zones/zone:zonename` をリフレッシュする必要があります。

すべてのゾーンを同じホスト `targethost` に退避するには、次のようにします。

```
global$ pfbash svccfg -s system/zones/zone
svc:/system/zones/zone> setprop evacuation/target=ssh://targethost
svc:/system/zones/zone> exit
global$ svcadm refresh svc:/system/zones
```

特定のゾーン `zonename` をターゲットホスト `targethost` に退避するには、ゾーン名をサービス名に追加します。

```
global$ pfbash svccfg -s system/zones/zone:zonename
svc:/system/zones/zone:zonename> setprop evacuation/target=ssh://targethost
svc:/system/zones/zone:zonename> exit
global$ svcadm refresh svc:/system/zones/zone:zonename
```

例 48 すべてのゾーンに対する退避ターゲットの設定とプロパティ値の確認

```
global$ pfbash svccfg -s system/zones/zone
svc:/system/zones/zone> setprop evacuation/target=ssh://global2
svc:/system/zones/zone> exit global$ svcprop -p "evacuation/target" svc:/system/zones/zone
ssh://global2
global$ svcadm refresh svc:/system/zones/zone
```

例 49 単一のカーネルゾーンに対する退避ターゲット設定

```
global$ pfbash svccfg -s system/zones/zone:kz1
svc:/system/zones/zone:kz1> setprop evacuation/target=ssh://global2
svc:/system/zones/zone:kz1> exit
global$ svcprop -p "evacuation/target" svc:/system/zones/zone:kz1
ssh://global2global$ svcadm refresh svc:/system/zones/zone:kz1
```

カーネルゾーン退避の準備としての保守モードの設定

ゾーン退避の準備を整えるには、保守モードを使用します。保守モードを開始すると、監査レコードがログに記録され、以降の接続、ブート、または任意のゾーンのシステムへの着信する移行が防止されます。ゾーンホストで管理タスクを実行する前に保守モードを開始すれば、ゾーン移行前にホストのサービスを停止できます。

保守の状態を変更するには、Maintenance and Repair 権利プロファイルが必要です。

ソースホストを保守モードに切り替え、オプションの関連メッセージを指定するには、次のコマンドを使用します。

```
sysadm maintain -s -m "message-text"
```

たとえば、システムをメッセージ付きで保守モードに切り替えたあと、保守のステータスを一覧表示するには、次のようにします。

```
global$ pfbash sysadm maintain -s -m "Updating system to new release"
global$ sysadm maintain -l
TYPE  USER   DATE           MESSAGE
admin root    2016-07-22 17:57  Updating system to new release
```

ステータスとして、保守のタイプ、`maintain` コマンドを実行したユーザー、実行時間、およびメッセージが表示されます。現時点でサポートされている保守のタイプは、`admin` だけです。

カーネルゾーンの退避

非対話型認証の設定、退避ターゲットの設定、および保守モードの開始が完了したら、ゾーンの退避に進むことができます。

デフォルトの `evacuate` コマンドは、実行中の各ゾーンを、そのゾーンの SMF サービスインスタンスの `evacuation/target` プロパティに構成された退避ターゲットにライブ移行しようとします。ある実行中のゾーンが `solaris-kz` 以外のブランドである場合、そのゾーンはライブ移行できないので、そのゾーンでは退避がスキップされます。

`-a` オプションを指定すれば、実行中でないゾーンも含め、すべてのゾーンを退避できます。実行中でないゾーンはコールド移行を使用して退避されます。

`-n` オプションを指定すれば、実行されない退避、つまり予行演習の退避を実行できます。移行が計画され、ターゲットホストへの予行演習の移行がゾーンごとに実行され、結果またはエラーが報告されます。

一部のゾーンが最初の実行で退避に失敗した場合のために、退避は複数回実行できるようになっています。戻されたステータスが成功になるのは、`sysadm evacuate` コマンドの実行後に、退避が全体として完了し、かつ実行中のゾーンがソースホスト上に存在していない場合だけです。

▼ 実行中のカーネルゾーンを退避する方法

始める前に 前のセクションで説明したように退避ターゲットが設定され、システムが保守モードになっていることを確認します。

1. カーネルゾーンを移行する権利が割り当てられた管理者になります。

詳細は、64 ページの「カーネルゾーン移行の実行に必要な権利」を参照してください。

2. 退避するゾーンが実行中であることを確認します。

次に出力例を示します。

```
source-host$ pfbash zoneadm list -cv
ID NAME          STATUS  PATH          BRAND  IP
0  global         running /            solaris shared
1  kzone          running -            solaris-kz excl
```

3. (オプション) 退避の予行演習を実行し、条件が適切に設定されていることを確認します。

次に出力例を示します。

```
source-host$ sysadm evacuate -n
sysadm: preparing zones for evacuation ... 1/1
sysadm: dry-run succeeded
```

4. ゾーンを退避します。

次に出力例を示します。

```
source-host$ sysadm evacuate -v
sysadm: preparing 3 zone(s) for evacuation ...
sysadm: initializing migration of kzone to new-host ...
...
sysadm: evacuation completed successfully.
```

例 50 成功するゾーン退避

```
root@ldom1-04:~# sysadm evacuate -v
sysadm: preparing 1 zone(s) for evacuation ...
sysadm: initializing migration of kzone1 to ldom1-08 ...
sysadm: evacuating 1 zone(s) ...
sysadm: migrating kzone1 to ldom1-08 ...
sysadm: evacuation completed successfully.
sysadm: kzone1: evacuated to ssh://ldom1-08
```

例 51 ゾーン退避時にスキップされるネイティブゾーン

この例では、退避時に 2 つのカーネルゾーンが正常にライブ移行され、1 つの solaris ブランドゾーンがスキップされているので、退避は全体としては「失敗」しています。ただし、カーネルゾーンは正常に退避されました。

```
root@global :~# sysadm evacuate -v
sysadm : preparing 3 zone(s) for evacuation ...
sysadm : initializing migration of kzone1 to global2 ...
sysadm : initializing migration of kzone2 to global2 ...
sysadm : evacuating 2 zone(s) ...
sysadm : migrating kzone1 to global2 ...
sysadm : migrating kzone2 to global2 ...
sysadm : evacuation failed .
sysadm : kzone1: evacuated to ssh://global2
sysadm : kzone2: evacuated to ssh://global2
```

```
sysadm : my-ngz : evacuation skipped: cannot evacuate solaris-brand zones
```

ゾーン退避のステータスの確認

退避のステータスを確認するには、`sysadm evacuate -l` コマンドを使用します。

```
root@global :~# sysadm evacuate -l
ZONENAME      STATE      DEST                                ERROR
kzone1        EVACUATED ssh://global2                      -
kzone2        EVACUATED ssh://global2                      -
my-ngz        SKIPPED
```

正常に退避されたゾーンの状態は、`EVACUATED` になります。

実行中の非大域ゾーンはライブ移行できないので、そのゾーンでは退避がスキップされます。このため、そのゾーンの状態は `SKIPPED` になります。

ある特定のゾーンの退避が失敗した場合、そのゾーンの状態は `FAILED` になり、`ERROR` の値によって詳細情報が提供されます。

カーネルゾーン退避後の保守モードの終了

管理タスクが完了し、必要に応じてリポートを行なったあと、保守モードをクリアするようにしてください。これは、退避したゾーンを戻すかどうかにかかわらず適用されます。保守モードが有効になっている間は、ゾーンをブートできません。

保守モードを終了するには、次のようにします。

```
root@global :~# sysadm maintain -e
```

退避されたカーネルゾーンを元のシステムに戻す

退避したゾーンを元のシステムに戻すには、ターゲットホストではなくソースホストで `sysadm evacuate` コマンドを実行します。戻すには `-r` オプションを指定します。

退避された各ゾーンがその宛先から移行されます。そのゾーンがそこでまだ実行されている場合は、元のソースホストに移行されます。実行中でないゾーンも戻す場合は、`-a` オプションを使用してすべてのゾーンを指定します。実行中でないゾーンはコールド移行されます。

```
source-host# sysadm evacuate -rv
```

例 52 退避されたゾーンを元のホストシステムに戻す

```

root@global:~# sysadm evacuate -rv
sysadm: preparing 2 zone(s) for return...
sysadm: initializing return of kzone1
sysadm: initializing return of kzone2
sysadm: returning 2 zone(s) ...
sysadm: migrating kzone1
sysadm: migrating kzone2
sysadm: return completed successfully.
sysadm: kzone1: returned
sysadm: kzone2: returned
root@global:~# sysadm evacuate -l
sysadm: no active evacuation
root@global:~# zoneadm list -cv
ID NAME          STATUS    PATH          BRAND    IP
  0 global         running  /             solaris  shared
  3 kzone2        running  -             solaris-kz  excl
  4 kzone1        running  -             solaris-kz  excl

```

カーネルゾーン退避の例

このセクションでは、2つのカーネルゾーンの退避に注釈を付けます。

例 53 カーネルゾーン退避の完全な手順

ゾーンの一覧を冗長出力で表示し、ステータスを確認します

```

root@global:~# zoneadm list -v
ID NAME          STATUS    PATH          BRAND    IP
  0 global         running  /             solaris  shared
 17 kzone2        running  -             solaris-kz  excl
 18 kzone1        running  -             solaris-kz  excl
 19 my-ngz        running  /system/zones/my-ngz  solaris  excl

```

ゾーンリスタートサービスの *evacuation/target SMF* プロパティを設定して移行ゾーンの宛先を指定し、ゾーンリスタートサービスをリフレッシュします

```

root@global:~# svccfg -s system/zones/zone
svc:/system/zones/zone> setprop evacuation/target=ssh://global2
svc:/system/zones/zone> exit
root@global:~# svcprop -p "evacuation/target" svc:/system/zones/zone
ssh://global2
root@global:~# svcadm refresh svc:/system/zones

```

システムをメッセージ付きで保守モードに切り替えます

```

root@global:~# sysadm maintain -s -m "Updating to new release"
root@global:~# sysadm maintain -l
TYPE  USER    DATE          MESSAGE
admin root    2016-03-16 17:57  Updating to new release

```

ゾーンを冗長出力付きで退避します

```

root@global:~# sysadm evacuate -v
sysadm : preparing 3 zone(s) for evacuation ...

```

```

sysadm : initializing migration of kzone1 to global2 ...
sysadm : initializing migration of kzone2 to global2 ...
sysadm : evacuating 2 zone(s) ...
sysadm : migrating kzone1 to global2 ...
sysadm : migrating kzone2 to global2 ...
sysadm : evacuation failed .
sysadm : kzone1: evacuated to ssh://global2
sysadm : kzone2: evacuated to ssh://global2
sysadm : my-ngz: evacuation skipped: cannot evacuate solaris-brand zones

```

退避されたゾーンを一覧表示します

```

root@global :~# sysadm evacuate -l
ZONENAME STATE      DEST      ERROR
kzone1    EVACUATED  ssh://global2  -
kzone2    EVACUATED  ssh://global2  -
my-ngz    SKIPPED    -             cannot evacuate solaris-brand zones

```

```

root@global :~# zoneadm -z my-ngz shutdown

```

システムの更新など、保守を実行します

```

root@global :~# pkg update ...
root@global :~# reboot ...

```

保守モードを終了します

```

root@global :~# sysadm maintain -e

```

退避したゾーンを戻します

```

root@global:~# sysadm evacuate -rv
sysadm: preparing 2 zone(s) for return...
sysadm: initializing return of kzone1
sysadm: initializing return of kzone2
sysadm: returning 2 zone(s) ...
sysadm: migrating kzone2
sysadm: migrating kzone1
sysadm: return completed successfully.
sysadm: kzone1: returned
sysadm: kzone2: returned

```

退避のステータスを確認します

```

root@global:~# sysadm evacuate -l
sysadm: no active evacuation

```

ゾーンの状態を確認します

```

root@global:~# zoneadm list -cv
ID NAME      STATUS  PATH      BRAND  IP
0  global      running /         solaris shared
3  kzone2      running -         solaris-kz excl
4  kzone1      running -         solaris-kz excl

```


Oracle Solaris カーネルゾーンの管理

この章で扱う内容は、次のとおりです。

- 97 ページの「カーネルゾーン環境での作業」
- 99 ページの「不変カーネルゾーンの操作」
- 99 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」
- 101 ページの「大域ゾーンから確認できるカーネルゾーンの補助状態」
- 102 ページの「カーネルゾーン内の非大域ゾーンの管理」
- 104 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」
- 106 ページの「カーネルゾーンのブートルoaderの起動」
- 108 ページの「NFS ストレージ URI とカーネルゾーン」
- 109 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、『[Creating and Using Oracle Solaris Zones](#)』の第4章、「[About Oracle Solaris Zones Administration](#)」を参照してください。カーネルゾーンは solaris10 ブランドゾーンをサポートしません。

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。

- 98 ページの「カーネルゾーンのプロセス情報の表示」
- 98 ページの「カーネルゾーンと大域ゾーンで重複するプロセス ID」
- 98 ページの「カーネルゾーンのゾーンパス」
- 98 ページの「カーネルゾーンでのリソース管理」

カーネルゾーンのプロセス情報の表示

大域ゾーンやカーネルゾーンホストシステムからカーネルゾーンのプロセスを直接表示することはできません。カーネルゾーンに関するプロセス情報を表示するには、次のように `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。

```
global$ pfbash zlogin -z kernel-zone process-management-command
```

たとえば次のコマンドは、カーネルゾーンホストシステム `global` から、カーネルゾーン `kzone1` 上の `syslogd` デーモンに関するプロセス情報を表示します。

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

カーネルゾーンと大域ゾーンで重複するプロセス 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、大域ゾーンでは `syslogd` を実行し、カーネルゾーンでは `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` コマンドに続いて `kill` コマンドを入力します。

```
global$ pfbash 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。

カーネルゾーンでのリソース管理

`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。

不変カーネルゾーンの操作

不変ゾーンでは、強制書き込みアクセス制御 (MWAC) が適用されるので、読み取り専用つまり不変のファイルシステムセキュリティーが提供されます。Oracle Solaris 11.4 では大域ゾーン、カーネルゾーン、および非大域ゾーンで不変ゾーンがサポートされます。詳細は、『[Creating and Using Oracle Solaris Zones](#)』の第 12 章、「[Configuring and Administering Immutable Zones](#)」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。手順の例については、[99 ページの「仮想 CD-ROM デバイスをカーネルゾーンに追加する方法」](#)を参照してください。

▼ 仮想 CD-ROM デバイスをカーネルゾーンに追加する方法

仮想 CD-ROM デバイスをカーネルゾーンに追加するには、このタスクを実行します。

1. ゾーン管理者になります。
詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」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。

```
global$ pfbash 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に追加します。

次に出力例を示します。

```
global$ zonecfg -z kzone  
zonecfg:kzone> add device  
zonecfg:kzone:device> set storage=dev:path-to-device  
zonecfg:kzone:device> end  
zonecfg:kzone> 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。

```
global$ zoneadm -z kzone reboot
```

5. カーネルゾーンにログインします。

```
global$ zlogin kzone
```

6. カーネルゾーンで、デバイスファイルシステム (devfs) を更新し、ハードウェア抽象化層 (hal) を再起動します。

このステップにより、hal サービスが仮想 CD-ROM デバイスを認識できるようになります。

```
kzone$ pfbash devfsadm -i zvblk
kzone$ svcadm restart hal
```

7. カーネルゾーンで、リムーバブルデバイスを一覧表示します。

次に出力例を示します。

```
kzone$ 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>
...

```

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、lofi デバイス /dev/lofi/1 への CD-ROM イメージ /root/sol-11_3-repo.full.iso 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. カーネルゾーンで、CD-ROM デバイスをマウントします。

```
kzone$ mount -F hsfs device-location /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。

```
kzone$ 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。

```
kzone$ eject cdrom
```

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

次に出力例を示します。

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

大域ゾーンから確認できるカーネルゾーンの補助状態

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、デバッグ操作やカーネル保守操作を開始する場合と、ゾーンを移行する場合だけです。

カーネルゾーンの補助状態は次のとおりです。

debugging

カーネルゾーンはカーネルデバッグ `kldb` 内にあります。カーネルゾーンは `running` 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。`kldb` 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、『[Creating and Using Oracle Solaris Zones](#)』の第3章、「[About Non-Global Zone Login](#)」を参照してください。

migrating-in

このゾーンはシステム上でブートされ、ライブ移行イメージを受信しています。ライブ移行が完了するまではまだ完全に実行されていません。

migrating-out

このゾーンは完全に実行されていますが、別のシステムにライブ移行されています。

no-config

このゾーンはシステムに認識されていますが、その構成がありません。ゾーンの状態は常に `incomplete` です。

panicked

ゾーンは `running` 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。`panicked` 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。

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、ゾーンに接続している必要があります。カーネルゾーンは、

ウォーム移行の前に `suspended` の補助状態として表示されます。第5章「Oracle Solaris カーネルゾーンの移行」を参照してください。

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。次に出力例を示します。

```
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の詳細は、[solaris-kz\(7\)](#) のマニュアルページを参照してください。

非大域ゾーンの状態については、『Introduction to Oracle Solaris Zones』の第1章、「Oracle Solaris Zones Introduction」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。

カーネルゾーン内の非大域ゾーンの管理

このセクションでは、カーネルゾーン内に入れ子になった非大域ゾーンの、ゾーン要件、MAC アドレス構成、およびゾーン管理上のその他の問題について説明します。

カーネルゾーン内のネイティブゾーンの要件

カーネルゾーンは、非大域ゾーン (具体的には `solaris` ゾーン) に対して大域ゾーンの役割を果たすことのできる、唯一のゾーンタイプです。カーネルゾーンの内側で、`solaris` ブランドの非大域ゾーンを作成、インストール、およびブートできます。カーネルゾーンを別のカーネルゾーンの内側に作成することはできません。カーネルゾーン内で実行されるゾーンは、入れ子になったゾーン、階層ゾーンなどと呼ばれることもあります。

カーネルゾーン内で実行される `solaris` ゾーンは、次の要件を満たす必要があります。

オペレーティングシステム

カーネルゾーンとその非大域ゾーンは少なくとも Oracle Solaris 11.2 を実行する必要があります。

Oracle Solaris 11 または Oracle Solaris 11.1 を実行している既存の `solaris` ゾーンは、少なくとも Oracle Solaris 11.2 に更新しないと、カーネルゾーン内で実行でき

ません。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 11.4](#)でのソフトウェアの追加とシステムの更新』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。

ネットワーク構成

カーネルゾーン内で実行される solaris ゾーンは排他的 IP ネットワークを使用する必要がありますので、[103 ページ](#)の「[複数の MAC アドレスをカーネルゾーンに追加する方法](#)」で示すように、カーネルゾーンを構成して追加の MAC アドレスを使用できるようにする必要があります。

システムリソース

カーネルゾーン内で実行されるゾーンは、カーネルゾーンで使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

非大域ゾーンを含むカーネルゾーンをクローニングする場合、外側のカーネルゾーンのみがクローニングされます。ゾーンクローニング処理では、カーネルゾーンの内側のゾーンはクローニングされません。[54 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

自動生成された 2 つの MAC アドレスをカーネルゾーンに追加するには、この手順を実行します。

1. ゾーン管理者になります。

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」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$ pfbash zonecfg -z kzone
zonecfg:kzone> add anet
zonecfg:kzone:anet> add mac
zonecfg:kzone:anet:mac> end
zonecfg:kzone:anet> add mac
zonecfg:kzone:anet:mac> end
zonecfg:kzone:anet> end
zonecfg:kzone> exit
```

3. 実行中のカーネルゾーンに変更を適用します。

```
global$ zoneadm -z kzone apply
zone 'kzone': Checking: Adding anet id=1
```

```
zone 'kzone': Applying the changes
```

別の方法として、ゾーンをブートして変更を適用することもできます。

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

```
global$ zlogin kzone
kzone$ pfbash 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   --
```

カーネルゾーン内での新しい solaris ゾーン構成

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

```
kzone$ pfbash zonecfg -z new-zone
Use 'create' to begin configuring a new zone.
zonecfg:new-zone> create -t SYSsolaris
zonecfg:new-zone> commit
zonecfg:new-zone> exit
kzone$ zoneadm -z new-zone install
kzone$ zoneadm -z new-zone boot
```

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

ホストデータからのストレージ情報

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判別されます。

ブートストレージが別のシステムで使用している場合、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$ pfbash 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。

暗号化鍵とホストデータ

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。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。ゾーンの暗号化鍵の回復が不可能な場合は、次のコマンドを実行して新しい暗号化鍵とホストデータを生成します。

```
$ pfbash zoneadm -z kernel-zone attach -x initialize-hostdata
```

ブートするには、カーネルゾーンで正しい鍵ソースが定義されている必要があります。`zoneadm migrate` で移行を行うと、ターゲットホストでまだゾーン構成が定義されていなかった場合、ソースホストからゾーン構成と鍵ソースのデータがコピーされます。

移行の前にターゲットホストでゾーン構成と鍵ソースを作成する必要がある場合は、ソースホストで `zonecfg export` コマンドを使用してファイルに情報をエクスポートし、ターゲットホストでその情報を使用して正しい鍵を含む構成を作成します。たとえば、`global1` から `global2` に移行するゾーンの構成を作成するには、`global1` で構成をネットワークパス上のファイルにエクスポートし、`global2` でそのファイルから構成を作成します。

```
global1$ zonecfg -z kzone1 export -f /net/example/path/kzone1.cfg
```

```
global2$ zonecfg -z kzone1 -f /net/example/path/kzone1.cfg
```

ターゲットホスト上のゾーンの既存の構成に対する鍵が正しくない場合、ゾーンへの接続やゾーンのブートを試みたときに、次のメッセージが表示されます。

```
zone 'kzone1': error: Encryption key is incorrect. See solaris-kz(7) for configuration migration
zone 'kzone1': procedure or update /etc/zones/keys/kzone1.
```

鍵が不正になる可能性があるのは、ゾーンを再インストールした場合や、鍵の再初期化を行う `-x initialize-hostdata` を指定してゾーンに接続した場合などです。

この問題を修正するには、ターゲットホストのゾーン構成を削除して構成をエクスポートし直すか、または構成を事前に作成しないでゾーンをターゲットホストに移行します。

カーネルゾーンのブートルoaderの起動

カーネルゾーンのブートルoaderは、カーネルゾーンのブート操作を管理します。ブートルoaderを起動するには、カーネルゾーンが `ready` または `installed` 状態になっている必要があります。

カーネルゾーンのブートルoader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ルoader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ルoader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。ブートルoader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、『[Creating and Using Oracle Solaris Zones](#)』の「[How to Log In to the Zone Console](#)」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 12 ブート環境の作成と管理](#)』の第1章、「[Introduction to Managing Boot Environments](#)」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 12 ブート環境の作成と管理](#)』の第2章、「[beadm Zones Support](#)」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. ゾーン管理者になります。

詳細は、『[Creating and Using Oracle Solaris Zones](#)』の「[Assigning Rights to Non-Root Users to Manage Zones](#)」を参照してください。

2. カーネルゾーンのコンソールにログインします。

```
global$ pfbash zlogin -C kzone
```

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。次に出力例を示します。

```
global$ pfbash zoneadm -z kzone boot -- -L
[Connected to zone 'kzone' 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。

```
global$ zoneadm -z kzone boot -- -Z boot-environment
```

例 54 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 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 55 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:

```

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 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、通常のパ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
```

URI 構文の例を次に示します。

```
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\(8\)](#) のマニュアルページおよび『[Creating and Using Oracle Solaris Zones](#)』の「[Managing Storage URIs and Shared Storage Resources](#)」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。

カーネルゾーン内でコアダンプを生成できない状況で、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。

追加情報については、[dumpadm\(8\)](#) および [coreadm\(8\)](#) のマニュアルページを参照してください。

索引

あ

暗号化

- 移行されるカーネルゾーンの鍵ソース, 105
- 移行時, 75

移行

- DAX と, 19
- iov プロパティ, 31
- pagesize-policy, 27
- RAD の要件, 15
- root 以外のユーザーによる, 64
- 移行後のカーネルゾーンの再開, 83
- ウォーム, 80
- ウォーム移行およびライブ移行の要件, 68
- 概要, 63
- コールド, 76
- 互換性要件, 71
- 準備, 69
- セキュア, 75
- ゾーン, 63
- 方式の選択, 66
- 要件, 67
- ライブ, 83
- 移行のための互換性, 71
- 一般的なゾーンの概念, 12
- 入れ子になったゾーン, 97
 - 参照 カーネルゾーン内で実行されているゾーン
- インストール
 - AI マニフェストによる方法, 46
 - sysconfig プロファイルによる方法, 49
 - インストールイメージによる方法, 51
 - インストールのログファイル, 44
 - 直接, 44
- 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, 51

ウォーム移行

- 定義, 63
- 要件, 68

か

カーネルゾーン

- AI マニフェストによるインストール, 46
- archiveadm を使用したクローニング, 55
- InfiniBand と anet リソースタイプ, 38
- NFS ストレージ URI, 108
- Oracle VM VirtualBox と, 12
- PCIe デバイスの共有, 31
- sysadm コマンド, 88
- sysconfig プロファイルによるインストール, 49
- SYSSolaris-kz-minimal テンプレート, 25
- SYSSolaris-kz テンプレート, 21
- アンインストール, 53
- 移行, 63
- 移行後の再開, 83
- 移行のセキュリティ保護, 75
- 移行のための互換性要件, 71
- 移行の要件, 67
- 移行方式, 66
- 一時停止と再開, 36
- インストール, 43
- インストールイメージによるインストール方法, 51
- インストール方法, 43
- ウォーム移行, 80
- 内側でネイティブゾーンを実行, 102
- 概要, 11
- 管理, 97
- 関連する概念, 12

- クローニング, 54
 - 権利プロファイル, 13
 - コアファイル, 109
 - 構成, 21
 - コールド移行, 76
 - 異なる OS または CPU への移行, 72
 - シャットダウン, 53
 - 重複するプロセス ID, 98
 - 使用することの利点, 11
 - ストレージデバイス, 28
 - ソフトウェアインシリコンと, 18
 - 大域ゾーンで表示される補助状態, 101
 - 大域ゾーンとの違い, 97
 - 退避, 87
 - 直接インストール, 44
 - 定義, 11
 - データベース, 28
 - 統合アーカイブ, 55
 - 統合アーカイブを使用したインストール, 55
 - ネットワークデバイス, 29
 - パフォーマンスのトラブルシューティング, 31
 - ブート順序, 28
 - ブートローダー, 106
 - 不変ゾーン, 99
 - プロセスの表示, 98
 - ペリファイドブート, 37
 - 変更不可能なゾーンパス, 98
 - ホスト情報, 104
 - メモリーと SYSsolaris-kz テンプレート, 26
 - メモリーの管理, 25
 - メモリーの予約, 17
 - 要件, 14
 - 要件の確認, 16
 - ライブ移行, 83
 - リソース管理, 98
 - リソースの管理, 22
 - リムーバブルデバイス, 99
 - ルートディスク, 26
 - カーネルゾーンのアニインストール, 53
 - カーネルゾーンの移行, 76
 - カーネルゾーンの一時的停止, 36
 - カーネルゾーンのインストール
 - Automated Installation (AI) マニフェスト, 46
 - sysconfig プロファイル, 49
 - インストールイメージ, 51
 - インストールイメージと AI マニフェスト, 52
 - コマンド行, 44
 - 直接, 44
 - カーネルゾーンのクローニング, 54
 - カーネルゾーンの再開, 36
 - 移行後, 83
 - カーネルゾーンのシャットダウン, 53
 - カーネルゾーンの停止, 53
 - カーネルゾーンのブートローダー, 106
 - カーネルゾーンのライブ移行, 83
 - 階層ゾーン, 97
 - 参照 カーネルゾーン内で実行されているゾーン
 - 鍵ソース
 - 移行されるカーネルゾーン, 105
 - 確認
 - カーネルゾーンでのホストのサポート, 16
 - 許可される MAC 接頭辞, 41
 - 許可される VLAN ID, 41
 - 仮想 LAN, 38
 - 動的な MAC アドレスと VLAN ID, 40
 - 管理 参照 構成
 - クロス CPU 移行
 - 準備, 72
 - 権利プロファイル
 - Maintenance and Repair, 89
 - カーネルゾーンの移行用, 64
 - ゾーン管理者向けの概要, 13
 - コアファイル, 109
 - 構成
 - カーネルゾーン, 21
 - カーネルゾーンのリソース, 22
 - 不変カーネルゾーン, 99
 - 読み取り専用カーネルゾーン, 99
 - コールド移行, 76
 - 定義, 63
- さ**
- シャドウ VNIC, 35
 - 重複するプロセス ID
 - カーネルゾーン, 98
 - 承認
 - カーネルゾーンの移行, 64
 - ストレージデバイス, 28

ストレージデバイスの追加, 28
 セキュアな移行, 75
 セキュリティー機能

- 暗号化された移行, 75
- 暗号化されたホストデータ, 104, 105
- 権利プロファイル, 13
- ベリファイドブート, 37
- ほかのゾーンと同様, 13

 ゾーン

- 移行, 63
- カーネルゾーン内で実行, 102
- 権利プロファイルと, 13
- ブランド 参照 `solaris-kz` ブランドゾーン

 ゾーンパス、変更不可能, 98
 ソフトウェアインシリコンの機能

- カーネルゾーンでの有効化, 18

 ソフトウェア要件, 14

た

退避

- `sysadm` コマンド, 88
- 移行後, 87
- 概要, 87
- ステータスの確認, 94
- ゾーンを元のシステムに戻す, 94
- ターゲットホストの設定, 90
- 方法, 92
- 保守モードの終了, 94
- 保守モードの設定, 91
- 要件, 89
- 例, 95

 直接インストール, 44
 データベース

- カーネルゾーンデータページ, 28

 統合アーカイブ

- クローニング, 54
- によるカーネルゾーンのインストール, 55

 トラブルシューティング

- 暗号化鍵が不正, 105
- データベースおよびページサイズ, 28
- 不十分なメモリー, 25
- ホストと仮想スイッチ間のトラフィック, 31

な

ネイティブゾーン, 97

- 参照 `solaris` ゾーン
- カーネルゾーンの内側で実行, 102
- カーネルゾーンの内側での要件, 102
- カーネルゾーンへの追加, 104

 ネットワークデバイス

- ID, 29
- 削除, 30
- 追加, 30

 ネットワークデバイスの追加, 29

は

ハードウェア要件, 14
 排他的 IP カーネルゾーン, 29
 ブート環境、指定, 106
 ブート順序

- 指定, 28

 ブートローダー, 106
 不変ゾーン, 99
 ブランドゾーン

- `solaris-kz`, 11

 プロセス ID

- カーネルゾーンでの可視性, 98

 ベリファイドブート, 37
 補助状態, 101
 ホストデータ, 104
 ホストの要件, 14
 ホスト ID, 104

ま

メモリー

- `pagesize-policy` プロパティ, 26
- 移行時に暗号化, 75
- カーネルゾーン用に予約, 17
- 管理, 25
- ページサイズ, 25

 メモリーの追加, 25

や
 要件

移行, 67
移行のための互換性, 71
ウォーム移行およびライブ移行, 68
読み取り専用カーネルゾーン, 99

ら

ライブ以降
Oracle Solaris 11.3 ターゲットシステムへ, 27
ライブ移行, 83
root 以外のユーザーによる, 64
定義, 64
要件, 68, 69
ライブゾーン再構成
手順, 57
リソース管理
カーネルゾーン, 98
リムーバブルデバイスの構成, 99
ルートディスクサイズの増量, 28
ログファイル
ゾーンのインストール, 44

A

add device コマンド, 28
ADI, 18
参照 SSM
anet リソース, 29
archiveadm コマンド, 54
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, 46

B

bootpri プロパティ, 28

C

capped-memory リソースタイプ, 25
cpu-arch リソースタイプ, 72
CPU
管理, 23

D

DAX, 18
dedicated-cpu リソースタイプ, 23

H

host-compatible プロパティ, 18
host-compatible リソースタイプ, 72

I

InfiniBand, 38
ioV プロパティ, 31

K

kz-migr サービス, 69

L

lofi デバイス, 99

M

MAC アドレス
静的, 39
動的, 40
Maintenance and Repair 権利プロファイル, 89

N

net リソース, 29
NFS ストレージ URI, 108

O

Oracle Solaris カーネルゾーン 参照 カーネルゾーン
Oracle VM VirtualBox
カーネルゾーン、および, 12

P

pagesize-policy プロパティ, 25, 26
PCIe デバイス
仮想マシン間での共有, 31

R

RAD
SMF サービス, 15

S

set_user_reserve.sh スクリプト, 17
Silicon Secured Memory, 11
参照 SSM
SMF サービス
kz-migr, 69
RAD, 15
solaris-kz ブランドゾーン, 11
solaris ゾーン
カーネルゾーンの内側での要件, 102
カーネルゾーンへの追加, 104
SPARC ファームウェア要件, 14
SR-IOV, 31
制限, 34
SR-IOV が有効になったカーネルゾーン
移行, 34
SSM, 18
suspend リソースタイプ, 36
sysadm evacuate コマンド, 89
sysadm maintain コマンド, 88
sysconfig プロファイル
カーネルゾーンのインストール, 49
SYSsolaris-kz-minimal テンプレート, 25
SYSsolaris-kz テンプレート, 21, 26

U

user_reserve_hint_pct チューニング可能, 17

V

VA マスク, 18

virtinfo コマンド, 16
virtual-cpu リソースタイプ
ゾーン構成における VCPU の削減, 59
追加, 24
VLAN 対応のカーネルゾーン
定義, 38
動的な MAC アドレスと VLAN ID, 40
VLAN タグ付け, 38

X

x86 BIOS の要件, 14

Z

ZFS ARC
キャッシュ要件, 14
チューニング, 17
Zone Cold Migration 権利プロファイル, 64
Zone Configuration 権利プロファイル, 64
Zone Migration 権利プロファイル, 64
zoneadm boot コマンド, 106
zoneadm clone コマンド, 54
zoneadm halt コマンド, 53
zoneadm install コマンド, 44
zoneadm list -s コマンド, 101
zoneadm migrate コマンド, 63, 70
zoneadm reboot コマンド, 53
zoneadm shutdown コマンド, 53
zoneadm suspend コマンド, 36
zoneadm uninstall コマンド, 53
zonecfg bwshare anet プロパティ, 36

