

Oracle® Solaris ゾーンの作成と使用

ORACLE®

Part No: E62804
2016年11月

Part No: E62804

Copyright © 2004, 2016, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

ドキュメントのアクセシビリティについて

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

Oracle Supportへのアクセス

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

目次

このドキュメントの使用法	15
1 非大域ゾーンの計画および構成方法	17
現在のシステム設定の評価	17
ディスク容量の要件	17
ゾーンサイズの制限	18
ゾーンのホスト名およびネットワーク要件の決定	18
ゾーンのホスト名	18
共有 IP ゾーンのネットワークアドレス	19
排他的 IP ゾーンのネットワークアドレス	20
ファイルシステムの構成	20
非大域ゾーン構成の作成、改訂、および削除	21
ゾーンを構成、検証、および確定する	22
▼ ゾーンの構成方法	22
次に進む手順	29
▼ 非大域ゾーンの構成を表示する方法	29
zonedcfg コマンドを使用してゾーン構成を変更する	29
▼ ゾーン構成内のリソースタイプを変更する方法	30
▼ ゾーン構成内のプロパティをクリアする方法	30
▼ zonedcfg コマンドを使用してゾーンの名前を変更する方法	31
▼ 専用のデバイスをゾーンに追加する方法	32
▼ 大域ゾーンの zone.cpu-shares を設定する方法	32
zonedcfg コマンドを使用してゾーン構成を元に戻す、または削除する	33
▼ ゾーン構成を元に戻す方法	33
▼ ゾーン構成を削除する方法	35
2 非大域ゾーンのインストール、停止処理、停止、アンインストール、クローニ	
ングについて	37
ゾーンのインストールと管理の概要	37
ゾーンの構築	38

ゾーンのインストール方法	40
zoneadmd デーモン	41
zsched ゾーンスケジューラ	42
ゾーンアプリケーション環境	42
ゾーンの停止処理、停止、リブート、およびアンインストールについて	43
ゾーンの停止処理	43
ゾーンの停止	43
ゾーンのリブート	44
ゾーンのブート引数	44
ゾーンの autoboot 設定	45
ゾーンのアンインストール	45
非大域ゾーンのクローニングについて	45
ゴールデンゾーンを作成する	47
3 非大域ゾーンのインストール、ブート、停止処理、停止、アンインストール、 およびクローニング	49
ゾーンのインストール	49
ゾーンのインストールとブート	50
▼ (オプション) インストール前に構成済みのゾーンを検証する方 法	50
▼ 構成済みのゾーンをインストールする方法	51
▼ インストールされた非大域ゾーンの UUID を取得する方法	53
▼ インストールした非大域ゾーンに不完全のマークを付ける方法	54
▼ (オプション) インストール済みのゾーンを準備完了状態に移行する 方法	55
▼ ゾーンのブート方法	56
▼ ゾーンをシングルユーザーモードでブートする方法	57
次に進む手順	57
ゾーンの停止処理、停止、リブート、およびアンインストール	57
▼ ゾーンを停止処理する方法	57
▼ ゾーンの停止方法	58
▼ ゾーンをリブートする方法	59
▼ zoneadm コマンドを使用してゾーンの名前を変更する方法	60
▼ ゾーンをアンインストールする方法	60
同一システム上での非大域ゾーンのクローニング	61
▼ ゾーンをクローンする方法	62
非大域ゾーンの移動	63
▼ 共有ストレージ上に存在しないゾーンを移動する方法	64
システムから非大域ゾーンを削除する	64

▼ 非大域ゾーンを削除する方法	64
4 非大域ゾーンへのログインについて	67
zlogin コマンド	67
ゾーンの内部構成	68
対話式システム構成ツール	69
ゾーン構成プロファイルの例	70
非大域ゾーンへのログイン方法	74
ゾーンコンソールログイン	74
ユーザーログインの方法	75
フェイルセーフモード	75
リモートログイン	75
対話型モードと非対話型モード	75
対話型モード	76
非対話型モード	76
5 非大域ゾーンへのログイン	77
初期のゾーンブートおよびゾーンログイン手順	77
ゾーンへのログイン	78
▼ 構成プロファイルを作成する方法	78
▼ ゾーンコンソールにログインして初期ゾーン構成を行う方法	79
▼ ゾーンコンソールへのログイン方法	80
▼ 対話型モードを使用してゾーンにアクセスする方法	80
▼ 非対話型モードを使用してゾーンにアクセスする方法	81
▼ 非大域ゾーンから抜ける方法	81
▼ フェイルセーフモードを使用してゾーンに入る方法	82
▼ zlogin を使用してゾーンを停止処理する方法	82
サービスの有効化	83
現在のゾーンの名前を出力する	83
6 ライブゾーン再構成	85
ライブゾーン再構成について	85
一時的な変更について	86
構成に対する変更について	86
構成への変更の適用について	86
ライブゾーン再構成の例	87
▼ 実行中のゾーンのライブ構成を検査する方法	87
▼ 予行演習を使用して可能な構成を表示する方法	87

▼ 永続的な構成変更を行い、変更を適用する方法	88
▼ 実行中のゾーンに anet リソースを一時的に追加する方法	88
▼ 実行中のゾーンに一時的な変更を加える方法	88
▼ 一時的な変更の確定中に障害から回復する方法	89
7 Oracle Solaris ゾーンの移行および変換	91
solaris ブランドゾーンの移行について	91
ゾーンの変換および Oracle Solaris ゾーンの変換について	92
ゾーンの変換のための Oracle Solaris 統合アーカイブの使用	93
ゾーンの変換のための従来のアーカイブの使用	93
zonep2vchk ツールを使用した大域から非大域への変換準備	93
solaris ブランドゾーンの移行およびゾーンの変換の要件	96
solaris ゾーンを移行するためのタスク	97
孤立したゾーンブート環境について	97
非大域ゾーンの別のマシンへの移行について	99
使用できないマシンからのゾーンの移行について	104
大域ゾーンの非大域ゾーンへの変換	105
8 ゾーンがインストールされている Oracle Solaris システムでの自動インストールおよびパッケージ	109
Oracle Solaris リリースを実行しているシステムのイメージパッケージングシステムソフトウェア	109
ゾーンのパッケージングの概要	109
パッケージとゾーンについて	111
ゾーンがインストールされているシステムへのパッケージの追加について	112
大域ゾーンでの pkg の使用	112
非大域ゾーンでの pkg install コマンドの使用	112
カスタム AI マニフェストを使用したゾーンへのパッケージの追加	112
ゾーン内でのパッケージの削除について	114
パッケージ情報の照会	114
インストール済みのゾーンがあるシステムでのプロキシ構成	114
大域ゾーンでのプロキシの構成	115
https_proxy および http_proxy を使用した system-repository プロキシのオーバーライド	115
ゾーンの並列更新	116
ゾーンの状態がパッケージの操作に与える影響	116

9 Oracle Solaris ゾーンの管理について	119
大域ゾーンの可視性とアクセス	120
ゾーン内でのプロセス ID の可視性	120
ゾーン内のシステム監視機能	121
zonestat ユーティリティーを使用したアクティブなゾーンの統計情報の報告	121
fsstat ユーティリティーを使用した非大域ゾーンのモニタリング	122
非大域ゾーンのノード名	122
ゾーンでの NFS サーバーの実行	123
ファイルシステムと非大域ゾーン	123
-o nosuid オプション	123
ゾーン内でのファイルシステムのマウント	124
ゾーン内でのファイルシステムのアンマウント	126
セキュリティの制限およびファイルシステムの動作	126
NFS クライアントとして機能する非大域ゾーン	129
ゾーン内での mknod の使用禁止	129
ファイルシステムの行き来	129
大域ゾーンから非大域ゾーンにアクセスする際の制限	130
共有 IP 非大域ゾーンにおけるネットワーク	131
共有 IP ゾーンの区分化	132
共有 IP ネットワークインタフェース	132
同一マシン上の共有 IP ゾーン間の IP トラフィック	133
共有 IP ゾーンでの Oracle Solaris IP フィルタ	133
共有 IP ゾーン内の IP ネットワークマルチパス	133
排他的 IP 非大域ゾーンにおけるネットワーク	134
排他的 IP ゾーンの区分化	135
排他的 IP データリンクインタフェース	135
同一マシン上の排他的 IP ゾーン間の IP トラフィック	135
排他的 IP ゾーンにおける Oracle Solaris IP フィルタ	136
排他的 IP ゾーン内の IP ネットワークマルチパス	136
非大域ゾーンでのデバイスの使用	136
/dev および /devices 名前空間	136
排他使用のデバイス	137
デバイスドライバの管理	137
非大域ゾーンで動作しないか、変更されるユーティリティー	138
非大域ゾーンでのアプリケーションの実行	139
非大域ゾーンで使用されるリソース制御	139
ゾーンがインストールされているシステムでの公平配分スケジューラ	140
大域ゾーンまたは非大域ゾーン内の FSS 配分分割	140

ゾーン間の配分均衡	140
ゾーンがインストールされているシステムでの拡張アカウントング	140
非大域ゾーン内の特権	141
ゾーン内での IP セキュリティーアーキテクチャーの使用	145
共有 IP ゾーン内の IP セキュリティーアーキテクチャー	145
排他的 IP ゾーン内の IP セキュリティーアーキテクチャー	146
ゾーン内での Oracle Solaris 監査の使用	146
ゾーン内のコアファイル	146
非大域ゾーン内での DTrace の実行	147
非大域 Oracle Solaris ゾーンでの時間値の設定	147
ゾーンがインストールされている Oracle Solaris システムのバックアップにつ いて	148
ループバックファイルシステムのディレクトリのバックアップ	148
大域ゾーンからのシステムのバックアップ	148
システム上の非大域ゾーンを個別にバックアップ	149
Oracle Solaris 統合アーカイブのバックアップの作成	149
非大域ゾーン内でバックアップするデータの決定	149
アプリケーションデータのみのバックアップ	149
一般的なデータベースバックアップ操作	150
テープによるバックアップ	150
非大域ゾーンの復元について	151
ゾーンがインストールされているシステムで使用するコマンド	151
10 Oracle Solaris ゾーンの管理	157
ppriv ユーティリティーの使用	157
▼ 大域ゾーンでの Oracle Solaris の特権を一覧表示する方法	158
▼ 非大域ゾーンの特権セットの表示方法	158
▼ 非大域ゾーンの特権セットを冗長出力で表示する方法	158
非大域ゾーンでの zonestat ユーティリティーの使用	159
▼ zonestat ユーティリティーを使用して、CPU およびメモリーの使 用効率のサマリーを表示する方法	160
▼ zonestat ユーティリティーを使用して、デフォルトの pset につい て報告する方法	160
▼ zonestat を使用した合計および最大使用効率の報告	161
▼ 排他的 IP ゾーンでのネットワーク帯域幅の使用効率を取得する方 法	161
すべてのゾーンのゾーン別の fstype 統計情報の報告	162
▼ -z オプションを使用して特定のゾーンのアクティビティーをモニ ターする方法	163

▼ すべてのゾーンのゾーン別の <code>fstype</code> 統計情報を表示する方法	163
非大域ゾーン内での DTrace の使用	163
▼ DTrace を使用する方法	164
非大域ゾーンの SMF サービスのステータスの確認	164
▼ コマンド行から SMF サービスのステータスを確認する方法	164
▼ ゾーン内から SMF サービスのステータスを確認する方法	165
稼働中の非大域ゾーン内でファイルシステムをマウントする	165
▼ LOFS を使用してファイルシステムをマウントする方法	166
▼ ZFS データセットを非大域ゾーンに委任する方法	166
大域ゾーン内の特定のファイルシステムへのアクセス権を非大域ゾーンに追加する	168
▼ 非大域ゾーンで CD または DVD メディアにアクセスする権限を追加する方法	168
ゾーンがインストールされている Oracle Solaris システムでの IP ネットワークマルチパスの使用	170
▼ 排他的 IP 非大域ゾーンで IP ネットワークマルチパスを使用する方法	170
▼ IP ネットワークマルチパス機能を共有 IP 非大域ゾーンに拡張する方法	170
排他的 IP 非大域ゾーンでのデータリンクの管理	171
▼ <code>dladm show-linkprop</code> の使用方法	172
▼ <code>dladm</code> を使用して、一時的なデータリンクを割り当てる方法	173
▼ <code>dladm reset-linkprop</code> の使用方法	174
ゾーンがインストールされている Oracle Solaris システムでの公平配分スケジューラの使用	174
▼ <code>prctl</code> コマンドを使用して大域ゾーンの FSS 配分を設定する方法	175
▼ ゾーンの <code>zone.cpu-shares</code> 値を動的に変更する方法	175
ゾーン管理での権利プロファイルの使用	176
▼ Zone Management プロファイルを割り当てる方法	176
ゾーンがインストールされている Oracle Solaris システムのバックアップ	176
▼ 統合アーカイブを使用した Oracle Solaris システムのバックアップ	176
▼ x64: ゾーン構成のコピーを出力する方法	177
非大域ゾーンの再作成	177
▼ 非大域ゾーンを個別に再作成する方法	177
11 不変ゾーンの構成と管理	179
読み取り専用ゾーンの概要	179

読み取り専用ゾーンの構成	180
zonecfg の file-mac profile プロパティ	180
zonecfg add dataset リソースポリシー	181
zonecfg add fs リソースポリシー	181
読み取り専用ゾーンの管理	182
zoneadm list -p の表示	182
書き込み可能なルートファイルシステムを持つ読み取り専用ゾーンを ブートするオプション	182
zlogin コマンドを使用したファイルの編集またはパッケージの追 加	183
不変大域ゾーン	183
不変大域ゾーンの構成	183
不変大域ゾーンの保守	184
12 Oracle Solaris ゾーンで発生するさまざまな問題のトラブルシューティ ング	185
排他的 IP ゾーンがデバイスを使用しているために dladm reset-linkprop が失敗する	185
ゾーン構成内に不正な特権セットが指定されている	185
ゾーンが停止しない	186
13 共有ストレージでの Oracle Solaris ゾーンの使用開始	187
ストレージ URI を使用した共有ストレージリソースについて	187
ローカルデバイス URI	187
論理ユニット URI	188
iSCSI URI	189
ストレージ URI と共有ストレージリソースの管理	190
Oracle Solaris ゾーン への共有ストレージリソースの割り当て	192
ゾーンの storage プロパティ	192
rootzpool リソース	193
zpool リソース	193
ゾーンの名前変更	194
ゾーン構成に関する制限事項	194
共有ストレージリソースでの Oracle Solaris ゾーン の自動 ZFS ストレージ プール管理	194
unavailable 状態について	195
追加の zoneadm サブコマンドオプション	196
ゾーンのインストール、クローニング、接続用のオプション	196
ゾーンのアンインストール用オプション	197

zoneadm コマンドの使用に関する制限事項	198
共有ストレージリソースでホストされる Oracle Solaris ゾーンの実装	198
共有ストレージ上のゾーンに関する zpoo1 構成の考慮事項	199
サンプルシナリオ	199
共有ストレージリソースでホストされる Oracle Solaris ゾーンの移行	210
共有ストレージゾーン構成に対する既存のゾーンの移動	212
▼ 共有ストレージ構成に既存のゾーンを移動する方法	212
▼ 共有ストレージ構成から既存のゾーンを移動する方法	213
▼ インストール済みゾーンに ZFS ストレージプールを追加する方 法	213
参考資料	214
マニュアルページ	214
Oracle Solaris 管理ガイド	215
用語集	217
索引	221

このドキュメントの使用方法

- **概要** – Oracle Solaris ゾーンの機能と関連リソース管理機能を設定および使用する方
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
- **前提知識** – Oracle Solaris 環境の管理経験。仮想化環境の経験も推奨されます。

製品ドキュメントライブラリ

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

フィードバック

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

非大域ゾーンの計画および構成方法

この章では、システムにゾーンを構成する前に実行する必要がある操作について説明します。また、ゾーンの構成方法、ゾーン構成の変更方法、およびシステムからゾーン構成を削除する方法についても説明します。

ゾーン構成プロセスの概要については、『Oracle Solaris ゾーン構成リソース』の第 1 章、「非大域ゾーンの構成」を参照してください。

solaris10 ブランドゾーンの構成については、『Oracle Solaris 10 ゾーンの作成と使用』を参照してください。

現在のシステム設定の評価

ゾーンは、Oracle Solaris 10 またはそれ以降のリリースが稼働する任意のマシンで使用できます。次に、ゾーンの使用に関連したマシンの主な考慮事項を示します。

- 各ゾーン内部で稼働するアプリケーションの性能要件。
- 各ゾーン内部で固有のファイルを保持するディスク容量がどれだけ利用可能か。

ディスク容量の要件

ゾーンが消費可能なディスク容量に関する制限はありません。大域管理者または適切な承認を持つユーザーが、容量の制限を担当します。大域管理者は、非大域ゾーンのルートファイルシステムを保持するのに十分なローカルストレージまたは共有ストレージがあることを確認する必要があります。小規模な単一プロセッサシステムでも、同時に稼働する多数のゾーンをサポートできます。

非大域ゾーンにインストールされるパッケージの特性は、ゾーンの容量要件に影響を与えます。また、パッケージの数も 1 つの要因です。

ディスク要件は、大域ゾーンに現在インストールされているパッケージとインストール済みのソフトウェアによって使用されるディスク容量によって決定されます。

1つのゾーンでは、ゾーンあたりに最低 150M バイトの空きディスク容量を必要とします。ただし、大域ゾーンが標準のすべての Oracle Solaris パッケージでインストールされている場合、必要な空きディスク容量は通常 500M バイトから 1G バイトです。さらに多くのソフトウェアが追加される場合、この数値は増える可能性があります。

ゾーンごとに 40M バイトの追加 RAM が推奨されていますが、十分なスワップ空間のあるマシンでは、これは必須ではありません。

ゾーンサイズの制限

ZFS データセットによって支援される `zonepath` を持つゾーンでは、ZFS データセットの割り当て制限を使用してゾーンサイズを制限できます。`zonepath` データセットにアクセスできる管理者は、データセットの `quota` および `reservation` プロパティを変更して、各ゾーンが消費できる最大のディスク容量を制御できます。これらのプロパティについては、[zfs\(1M\)](#) のマニュアルページで説明されています。

また、管理者は、ZFS ボリュームを固定サイズで作成して、そのボリュームのデータセット内にゾーンをインストールすることもできます。このボリュームは、その内部にインストールされたゾーンのサイズを制限します。

ゾーンのホスト名およびネットワーク要件の決定

ゾーンのホスト名を決定する必要があります。

排他的 IP ゾーンの内部で、大域ゾーンに対して行なったようにアドレスを構成します。

ネットワーク接続を持つ共有 IP ゾーンに対して、次のいずれかを実行する必要があります。

- ゾーンに IPv4 アドレスを割り当てます
- ゾーンに IPv6 アドレスを構成して割り当てます

排他的 IP タイプと共有 IP タイプの詳細は、『[Oracle Solaris ゾーン構成リソース](#)』の「[ゾーンネットワークインタフェース](#)」を参照してください。

ゾーンのホスト名

NIS または DNS ネームサービス、あるいは LDAP ディレクトリサービスを使用している場合は、サーバー上のデータベース (`hosts.byname` など) にホスト情報が格納されます。

ネームサービス用にローカルファイルを使用する場合は、`/etc/inet/hosts` ファイル内で `hosts` データベースが保持されます。ゾーンネットワークインタフェースのホスト名の解決は、`/etc/inet/hosts` 内のローカル `hosts` データベースで行われます。あるいは、共有 IP ゾーンの場合、ゾーンの構成時に IP アドレス自体を直接指定することで、ホスト名の解決を不要にできます。詳細は、[hosts\(4\)](#) および [nodename\(4\)](#) のマニュアルページを参照してください。『Oracle Solaris 11.3 でのネットワークコンポーネントの構成と管理』の第 3 章、「Oracle Solaris での IP インタフェースとアドレスの構成および管理」も参照してください。

共有 IP ゾーンのネットワークアドレス

ネットワーク接続を必要とする各共有 IP ゾーンには、1 つ以上の一意の IP アドレスが与えられます。IPv4 と IPv6 の両方のアドレスがサポートされます。

IPv4 のゾーンネットワークアドレス

IPv4 を使用している場合、アドレスを取得してゾーンに割り当てます。アドレスをゾーンに割り当てるときは、CIDR 表記 (`192.168.1.1/24` など) を使用してアドレスを指定できます。

共有 IP ゾーンの場合は、ゾーンの構成時に IP アドレス自体を直接指定することで、ホスト名の解決を不要にできます。

詳細は、[hosts\(4\)](#)、[netmasks\(4\)](#)、および [nodename\(4\)](#) を参照してください。

IPv6 のゾーンネットワークアドレス

IPv6 を使用している場合、アドレスを手動で構成する必要があります。通常、次の 2 種類のアドレスを最小限構成する必要があります。

リンクローカルアドレス	リンクローカルアドレスの書式は、 <code>fe80::64</code> ビットインタフェース <code>ID/10</code> です。 <code>/10</code> は、接頭辞の長さが 10 ビットであることを示します。
-------------	---

グローバルユニキャストアドレス	大域ユニキャストアドレスは、管理者がサブネットごとに構成した 64 ビット接頭辞および 64 ビットのインタフェース ID に基づきます。接頭辞は、IPv6 を使用するよう構成されている同一サブネット上の任意のシステムで、 <code>ipadm show-addr</code> コマンドを実行して取得できます。
-----------------	---

通常、64 ビットのインタフェース ID は、システムの MAC アドレスから取得されます。ゾーンで使用するために、次の規則を使

用して、大域ゾーンの IPv4 アドレスから一意の代替アドレスを取得できます。

16 ビットのゼロ:IPv4 アドレスの上位 16 ビット:IPv4 アドレスの下位 16 ビット:ゾーンで一意の番号

大域ゾーンの IPv4 アドレスは 192.168.200.10 と想定しています。このアドレスは次のように 16 進数に変換されます。

- 192 = c0
- 168 = a8
- 200 = c8
- 10 = 0a

したがって、ゾーン固有の番号 1 を使用した非大域ゾーンの適正なリンクローカルアドレスは `fe80::c0a8:c80a:1/10` になります。

そのサブネットで使用中の大域接頭辞が `2001:0db8:aabb:ccdd/64` である場合、同じ非大域ゾーン用の一意の大域ユニキャストアドレスは `2001:0db8:aabb:ccdd::c0a8:c80a:1/64` です。IPv6 アドレスを構成する際、接頭辞の長さを指定する必要があることに注意してください。

リンクローカルおよび大域ユニキャストアドレスの詳細は、[ipadm\(1M\)](#) および [inet6\(7P\)](#) のマニュアルページを参照してください。

排他的 IP ゾーンのネットワークアドレス

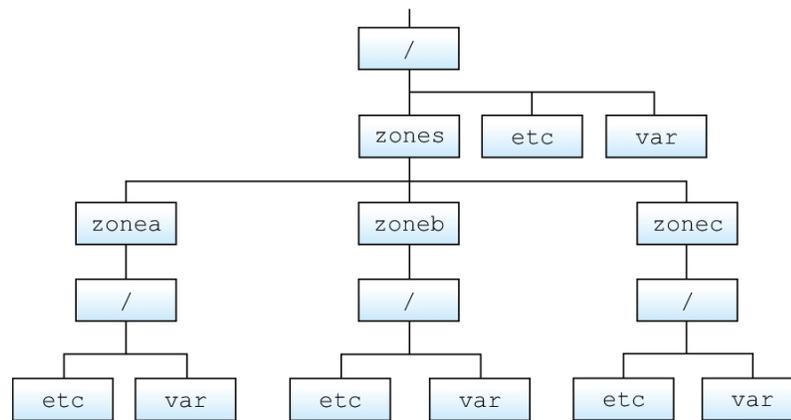
排他的 IP ゾーンの内部で、大域ゾーンと同様の方法でアドレスを構成します。DHCP および IPv6 ステータスアドレスの自動構成を使用してアドレスを構成することもできます。IP アドレス構成については、『[Oracle Solaris 11.3 でのネットワークコンポーネントの構成と管理](#)』の第 3 章、「[Oracle Solaris での IP インタフェースとアドレスの構成および管理](#)」を参照してください。

ファイルシステムの構成

仮想プラットフォームを設定する際、実行するマウントを多数指定できます。ループバック仮想ファイルシステム (LOFS) を使用してファイルシステムをゾーンにループバックマウントする場合、`nodevices` オプションを指定してファイルシステムをマウントする必要があります。`nodevices` オプションの詳細は、[123 ページの「ファイルシステムと非大域ゾーン」](#)を参照してください。

LOFS を使用すると、代替パス名を使用してファイルにアクセスできるように、新しい仮想ファイルシステムを作成できます。非大域ゾーンでは、ループバックマウントにより、ファイルシステム階層がゾーンのルート下に複製されているように見えます。ゾーン内では、ゾーンのルートから始まるパス名を使ってすべてのファイルにアクセスできるようになります。LOFS マウントでは、ファイルシステムの名前空間が維持されます。

図 1 ループバックマウントされたファイルシステム



詳細は、lofs(7S) のマニュアルページを参照してください。

非大域ゾーン構成の作成、改訂、および削除

タスク	説明	手順
非大域ゾーンを構成します。	<p>zonecfg コマンドを使用してゾーンの作成、構成の検証および確定を行います。スクリプトを使用して、システム上の複数のゾーンを構成およびブートすることもできます。</p> <p>非大域ゾーンの構成を表示するには、zonecfg コマンドを使用できます。</p>	22 ページの「ゾーンを構成、検証、および確定する」
ゾーン構成を変更します。	これらの手順を使用して、ゾーンの構成内のリソースタイプを変更するか、ゾーンの名前など	29 ページの「zonecfg コマンドを使用してゾーン構成を変更する」

タスク	説明	手順
	のプロパティタイプを変更するか、またはゾーンに専用のデバイスを追加します。	
ゾーン構成を元に戻すか、ゾーン構成を削除します。	zonecfg コマンドと revert サブコマンドを使用して、ゾーン構成に対して行なったリソース設定を取り消すか、ゾーン構成を削除します。	33 ページの「zonecfg コマンドを使用してゾーン構成を元に戻す、または削除する」
ゾーン構成を削除します。	zonecfg コマンドと delete サブコマンドを使用して、システムからゾーン構成を削除します。	35 ページの「ゾーン構成を削除する方法」

ゾーンを構成、検証、および確定する

次のアクションを実行するには、zonecfg コマンド (zonecfg(1M) のマニュアルページに記載) を使用します。

- ゾーン構成を作成します。
- 必要な情報がすべて存在することを検証します。
- 非大域ゾーン構成を確定します。

zonecfg コマンドは、大域ゾーンのリソース管理設定を永続的に指定する場合にも使用できます。

zonecfg ユーティリティーを使用してゾーンを構成する際、revert サブコマンドを使用して、リソースの設定を元に戻すことができます。33 ページの「ゾーン構成を元に戻す方法」を参照してください。

非大域ゾーンの構成を表示する方法については、29 ページの「非大域ゾーンの構成を表示する方法」を参照してください。

▼ ゾーン構成方法

非大域ゾーンの作成に必須の要素は、rootzpool リソースを伴うゾーンの場合、zonename および zonepath プロパティだけです。そのほかのリソースおよびプロパティはオプションです。オプションのリソースには、dedicated-cpu リソースと capped-cpu リソースのどちらを使用するかを決めるなど、選択肢の中から選ぶ必要があるものもあります。使用可能な zonecfg プロパティおよびリソースについては、『Oracle Solaris ゾーン構成リソース』の「ゾーン構成データ」を参照してください。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. 選択したゾーン名を使用して、ゾーン構成を設定します。

この手順例では、my-zone という名前を使用します。

```
global# zonecfg -z my-zone
```

このゾーンの初回構成時には、次のシステムメッセージが表示されます。

```
Use 'create' to begin configuring a new zone.
```

3. 新しいゾーン構成を作成します。

この手順では、デフォルト設定を使用します。

```
zonecfg:my-zone> create
create: Using system default template 'SYSdefault'
```

```
zonecfg:my-zone> info
```

```
zonename: my-zone>
zonepath.template: /system/zones/{zonename}
zonepath: /system/zones/my-zone
...
```

注記 - `set zonepath=` を使用してゾーンのパスを選択する場合、そのゾーンは ZFS データセット上に存在する必要があります。ZFS データセットは、ゾーンのインストール時または接続時に自動的に作成されます。ZFS データセットを作成できない場合、ゾーンのインストールや接続は行われません。ゾーンのパスの親ディレクトリが存在する場合、その親ディレクトリはマウントされるデータセットのマウントポイントである必要があります。

4. autoboot 値を設定します。

true に設定すると、大域ゾーンのブート時にこのゾーンが自動的にブートします。デフォルト値は false です。ゾーンを自動的にブートするには、ゾーンサービス `svc:/system/zones:default` も有効になっている必要があることに注意してください。このサービスは、デフォルトで有効になっています。

```
zonecfg:my-zone> set autoboot=true
```

5. ゾーンの持続的なブート引数を設定します。

```
zonecfg:my-zone> set bootargs="-m verbose"
```

6. 1 つの CPU をこのゾーン専用割り当てます。

```
zonecfg:my-zone> add dedicated-cpu
```

a. CPU の数を設定します。

```
zonecfg:my-zone:dedicated-cpu> set ncpus=1-2
```

b. (オプション) 重要性を設定します。

```
zonecfg:my-zone:dedicated-cpu> set importance=10
```

デフォルト値は 1 です。

c. 指定を終了します。

```
zonecfg:my-zone:dedicated-cpu> end
```

7. 権限のデフォルトセットを修正します。

```
zonecfg:my-zone> set limitpriv="default,sys_time"
```

この行は、システムクロックを設定する機能をデフォルトの特権セットに追加します。

8. スケジューリングクラスを FSS に設定します。

```
zonecfg:my-zone> set scheduling-class=FSS
```

9. メモリー上限を追加します。

```
zonecfg:my-zone> add capped-memory
```

a. メモリー上限を設定します。

```
zonecfg:my-zone:capped-memory> set physical=1g
```

b. スワップメモリーの上限を設定します。

```
zonecfg:my-zone:capped-memory> set swap=2g
```

c. ロックされたメモリーの上限を設定します。

```
zonecfg:my-zone:capped-memory> set locked=500m
```

d. メモリー上限の指定を終了します。

```
zonecfg:my-zone:capped-memory> end
```

注記 - capped-memory リソースを使用するには、resource-cap パッケージが大域ゾーン内にインストールされている必要があります。

10. ファイルシステムを追加します。

```
zonecfg:my-zone> add fs
```

- a. ファイルシステムのマウントポイント (この手順では `/usr/local`) を設定します。

```
zonecfg:my-zone:fs> set dir=/usr/local
```

- b. 大域ゾーン内の `/opt/local` を、構成中のゾーン内で `/usr/local` としてマウントすることを指定します。

```
zonecfg:my-zone:fs> set special=/opt/local
```

非大域ゾーン内では、`/usr/local` ファイルシステムは読み取りおよび書き込みが可能です。

- c. ファイルシステムのタイプ (この手順では `lofs`) を指定します。

```
zonecfg:my-zone:fs> set type=lofs
```

このタイプは、カーネルとそのファイルシステムとの相互動作の方法を示します。

- d. ファイルシステムの指定を終了します。

```
zonecfg:my-zone:fs> end
```

この手順を複数回実行することで、複数のファイルシステムを追加できます。

11. 必要に応じて `hostid` を設定します。

```
zonecfg:my-zone> set hostid=80f0c086
```

12. ストレージプール `tank` の `sales` という ZFS データセットを追加します。

```
zonecfg:my-zone> add dataset
```

- a. ZFS データセット `sales` のパスを指定します。

```
zonecfg:my-zone> set name=tank/sales
```

- b. データセットの指定を終了します。

```
zonecfg:my-zone> end
```

データセットリソースとともに使用すべき唯一のデータセットタイプは ZFS ファイルシステムです。ゾーン管理者は、子ファイルシステムとその子孫のクローンを作成できます。ゾーン管理者は、データセットのプロパティを変更したり、圧縮や暗号化を制御したりできます。

13. 自動 VNIC を持つ排他的 IP ゾーンを作成します。

```
zonecfg:my-zone> set ip-type=exclusive
```

```
zonecfg:my-zone> add anet
```

- a. 作成されるリンクの基礎となるリンクとして **auto** を指定します。

```
zonecfg:my-zone:anet> set lower-link=auto
```

zoneadmd デーモンは、ゾーンがブートするたびに VNIC が作成されるリンクを自動的に選択します。データリンクを選択するときは、IPoIB リンクはスキップされます。

- b. 指定を終了します。

```
zonecfg:my-zone:anet> end
```

14. デバイスを追加します。

```
zonecfg:my-zone> add device
```

- a. デバイスの一致 (この手順では **/dev/sound/***) を設定します。

```
zonecfg:my-zone:device> set match=/dev/sound/*
```

- b. デバイスの指定を終了します。

```
zonecfg:my-zone:device> end
```

この手順を複数回実行することで、複数のデバイスを追加できます。

15. **IB 診断ツール以外の OFUV (Open Fabrics User Verbs) のコンポーネントに対する OFUV デバイスを追加します。**

```
zonecfg:my-zone> add device
```

- a. デバイスの一致 (この手順では **infiniband/ofs/***) を設定します。

```
zonecfg:my-zone:device> set match=infiniband/ofs/*
```

- b. デバイスの指定を終了します。

```
zonecfg:my-zone:device> end
```

IB 診断ツールは、非大域ゾーンではサポートされていません。追加されたデバイスは、OFUV のコンポーネント (Verb や rdma_cm など) とともに使用できます。

この手順を複数回実行することで、複数のデバイスを追加できます。

16. **IB 診断ツール以外の OFUV のコンポーネントに対する OFUV デバイスを追加します。**

```
zonecfg:my-zone> add device
```

- a. デバイスの一致 (この手順では **infiniband/hca/***) を設定します。

```
zonecfg:my-zone:device> set match=infiniband/hca/*
```

- b. デバイスの指定を終了します。

```
zonecfg:my-zone:device> end
```

IB 診断ツールは、非大域ゾーンではサポートされていません。追加されたデバイスは、OFUV のコンポーネント (Verb や rdma_cm など) とともに使用できます。

この手順を複数回実行することで、複数のデバイスを追加できます。

17. **format** コマンドによるディスクのラベル付けを可能にするには、ディスクおよび LUN 全体をゾーンに委任し、また **allow-partition** プロパティを設定している必要があります。

```
zonecfg:my-zone> add device
```

- a. この手順で、デバイスの一致 **/dev/*dsk/c2t40d3*** を設定します。

```
zonecfg:my-zone:device> set match=/dev/*dsk/c2t40d3*
```

- b. **allow-partition** を **true** になるように設定します。

```
zonecfg:my-zone:device> set allow-partition=true
```

- c. デバイスの指定を終了します。

```
zonecfg:my-zone:device> end
```

この手順を複数回実行することで、複数のデバイスを追加できます。

18. ディスクに対して **uscsi** 操作を可能にするには、**allow-raw-io** プロパティを設定している必要があります。

```
zonecfg:my-zone> add device
```

- a. この手順で、デバイスの一致 **/dev/*dsk/c2t40d3*** を設定します。

```
zonecfg:my-zone:device> set match=/dev/*dsk/c2t40d3*
```

- b. **allow-raw-io** を **true** になるように設定します。

```
zonecfg:my-zone:device> set allow-raw-io=true
```

- c. デバイスの指定を終了します。

```
zonecfg:my-zone:device> end
```



注意 - ゾーンでディスクに対する `uscsi` 操作の実行を可能にすると、ゾーンはディスクと同じバスに接続されている、すべての他のデバイスにもアクセスできます。したがって、この機能を有効にするとセキュリティリスクが生じ、同じバス上のリソースを使用する大域ゾーンまたはほかのゾーンに対する攻撃が可能になります。詳細は、[uscsi\(7I\)](#)を参照してください。

この手順を複数回実行することで、複数のデバイスを追加できます。

19. プロパティ名を使用して、ゾーン規模のリソース制御を追加します。

```
zonecfg:my-zone> set max-sem-ids=10485200
```

この手順を複数回実行することで、複数のリソース制御を追加できます。

20. リソースタイプ `attr` を使用してコメントを追加します。

```
zonecfg:my-zone> add attr
```

a. 名前を `comment` に設定します。

```
zonecfg:my-zone:attr> set name=comment
```

b. タイプを `string` に設定します。

```
zonecfg:my-zone:attr> set type=string
```

c. 値をゾーンの内容を示すコメントに設定します。

```
zonecfg:my-zone:attr> set value="This is my work zone."
```

d. リソースタイプ `attr` の指定を終了します。

```
zonecfg:my-zone:attr> end
```

21. ゾーンの構成を検証します。

```
zonecfg:my-zone> verify
```

22. ゾーンの構成を確定します。

```
zonecfg:my-zone> commit
```

23. `zonecfg` コマンドを終了します。

```
zonecfg:my-zone> exit
```

プロンプトで `commit` コマンドを明示的に入力しなくても、`exit` を入力するか EOF が発生すると、`commit` の実行が自動的に試みられます。

コマンド行での複数のサブコマンドの使用

ヒント - `zonecfg` コマンドは、複数のサブコマンドもサポートします。次に示すように、同じシェル呼び出しで引用符で囲み、セミコロンで区切ります。

```
global# zonecfg -z my-zone "create ; set zonepath=/zones/my-zone"
```

共有 IP ゾーンの場合、`zonecfg net` リソースで静的アドレスを割り当てできるだけです。このアドレスは、コマンド行で指定することはできません。

次に進む手順

確定済みのゾーン構成をインストールする方法については、[50 ページの「ゾーンのインストールとブート」](#)を参照してください。

▼ 非大域ゾーンの構成を表示する方法

大域ゾーン内の大域管理者か、またはこの手順を実行するための正しい権利プロファイルを持つユーザーである必要があります。

1. 管理者になります。

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

2. ゾーンの構成を表示します。

```
global# zonecfg -z zonename info
```

zonecfg コマンドを使用してゾーン構成を変更する

`zonecfg` コマンドを使用して、次の操作を実行することもできます。

- ゾーン構成内のリソースタイプを変更します
- ゾーン構成内のプロパティの値をクリアします
- 複製したデバイスをゾーンに追加します
- ゾーンの特権セットを変更します
- ストレージを追加および削除します

▼ ゾーン構成内のリソースタイプを変更する方法

リソースタイプを選択して、そのリソースの仕様を変更できます。

大域ゾーン内の大域管理者か、またはこの手順を実行するための正しい権利プロファイルを持つユーザーである必要があります。

1. 管理者になります。

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

2. 変更するゾーン (この手順では `my-zone`) を選択します。

```
global# zonecfg -z my-zone
```

3. 変更するリソースタイプ (リソース制御など) を選択します。

```
zonecfg:my-zone> select rctl name=zone.cpu-shares
```

4. 現在の値を削除します。

```
zonecfg:my-zone:rctl> remove value (priv=privileged,limit=20,action=none)
```

5. 新しい値を追加します。

```
zonecfg:my-zone:rctl> add value (priv=privileged,limit=10,action=none)
```

6. 改定された `rctl` の指定を終了します。

```
zonecfg:my-zone:rctl> end
```

7. ゾーンの構成を確定します。

```
zonecfg:my-zone> commit
```

8. `zonecfg` コマンドを終了します。

```
zonecfg:my-zone> exit
```

プロンプトで `commit` コマンドを明示的に入力しなくても、`exit` を入力するか EOF が発生すると、`commit` の実行が自動的に試みられます。

`zonecfg` で行なった確定済みの変更は、ゾーンの次回ブート時に適用されます。

▼ ゾーン構成内のプロパティをクリアする方法

スタンドアロンのプロパティをリセットするには、この手順を使用します。

1. 管理者になります。

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

2. 変更するゾーン (この手順では `my-zone`) を選択します。

```
global# zonecfg -z my-zone
```

3. 変更対象のプロパティー (この手順では既存のプールの関連付け) をクリアーします。

```
zonecfg:my-zone> clear pool
```

4. ゾーン構成を確定します。

```
zonecfg:my-zone> commit
```

5. `zonecfg` コマンドを終了します。

```
zonecfg:my-zone> exit
```

プロンプトで `commit` コマンドを明示的に入力しなくても、`exit` を入力するか EOF が発生すると、`commit` の実行が自動的に試みられます。

`zonecfg` で行なった確定済みの変更は、ゾーンの次回ブート時に適用されます。

▼ zonecfg コマンドを使用してゾーンの名前を変更する方法

構成済み状態またはインストール済み状態にあるゾーンの名前を変更するには、この手順を使用します。

`zonename` は既存の `zpool` 名の一部であるため、`rootzpool` または `zpool` リソースを含むインストール済み状態のゾーンの名前は変更できません。これらのゾーンの名前を変更するには、この手順の最後の「共有ストレージ上のゾーンの名前変更」を参照してください。

大域ゾーン内の大域管理者か、またはこの手順を実行するための正しい権利プロファイルを持つユーザーである必要があります。

1. 管理者になります。

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

2. 名前を変更するゾーン (この手順では `my-zone`) を選択します。

```
global# zonecfg -z my-zone
```

3. ゾーンの名前を `newzone` などに変更します。

```
zonecfg:my-zone> set zonename=newzone
```

4. 変更を確定します。

```
zonecfg:newzone> commit
```

5. zonecfg コマンドを終了します。

```
zonecfg:newzone> exit
```

zonecfg で行なった確定済みの変更は、ゾーンの次回ブート時に適用されます。

▼ 専用のデバイスをゾーンに追加する方法

走査デバイスを非大域ゾーン構成内に配置するには、次の指定を使用します。

大域ゾーン内の大域管理者か、またはこの手順を実行するための適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. デバイスを追加します。

```
zonecfg:my-zone> add device
```

3. デバイスの一致 (この手順では /dev/scsi/scanner/c3t4*) を設定します。

```
zonecfg:my-zone:device> set match=/dev/scsi/scanner/c3t4*
```

4. デバイスの指定を終了します。

```
zonecfg:my-zone:device> end
```

5. zonecfg コマンドを終了します。

```
zonecfg:my-zone> exit
```

▼ 大域ゾーンの zone.cpu-shares を設定する方法

この手順を使用して、大域ゾーンの配分を永続的に設定します。

大域ゾーン内の大域管理者か、またはこの手順を実行するための正しい権利プロファイルを持つ大域ゾーン内のユーザーである必要があります。

1. 管理者になります。

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

2. zonecfg コマンドを使用します。

```
# zonecfg -z global
```

3. 大域ゾーンの 5 つの共有を設定します。

```
zonecfg:global> set cpu-shares=5
```

4. zonecfg を終了します。

```
zonecfg:global> exit
```

zonecfg コマンドを使用してゾーン構成を元に戻す、または削除する

zonecfg(1M) のマニュアルページの記述に従って、zonecfg コマンドを使用し、ゾーンの構成を元に戻すか、またはゾーン構成を削除します。

▼ ゾーン構成を元に戻す方法

zonecfg ユーティリティーによるゾーンの構成中にゾーン構成に対して行なったりリソース設定を取り消すには、revert サブコマンドを使用します。

大域ゾーン内の大域管理者か、またはこの手順を実行するためのゾーンのセキュリティー権利プロファイルを持つ大域ゾーン内のユーザーである必要があります。

1. 管理者になります。

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

2. tmp-zone という名前のゾーンを構成中に、info と入力して構成を表示します。

```
zonecfg:tmp-zone> info
```

構成の net リソースセグメントが、次のように表示されます。

```
·
·
·
```

```
fs:
    dir: /tmp
    special: swap
    type: tmpfs
net:
    address: 192.168.0.1
    physical: eri0
device
    match: /dev/pts/*
.
.
.
```

3. ネットアドレスを削除します。

```
zonecfg:tmp-zone> remove net address=192.168.0.1
```

4. net エントリが削除されたことを確認します。

```
zonecfg:tmp-zone> info
.
.
.
fs:
    dir: /tmp
    special: swap
    type: tmpfs
device
    match: /dev/pts/*
.
.
.
```

5. revert と入力します。

```
zonecfg:tmp-zone> revert
```

6. 次の質問に **yes** で応答します。

```
Are you sure you want to revert (y/[n])? y
```

7. ネットアドレスが再び存在することを確認します。

```
zonecfg:tmp-zone> info
.
.
.
fs:
    dir: /tmp
    special: swap
    type: tmpfs
net:
    address: 192.168.0.1
    physical: eri0
device
    match: /dev/pts/*
.
.
.
```

▼ ゾーン構成を削除する方法

zonecfg と delete サブコマンドを使用して、システムからゾーン構成を削除します。

大域管理者か、またはこの手順を実行するためのセキュリティー権利プロファイルを持つ大域ゾーン内のユーザーである必要があります。

1. 管理者になります。

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

2. 次の 2 つの方法のいずれかを使用して、ゾーン a-zone のゾーン構成を削除します。

- -F オプションを使用して操作を強制実行します。

```
global# zonecfg -z a-zone delete -F
```

- システムプロンプトに yes で応答し、対話的にゾーンを削除します。

```
global# zonecfg -z a-zone delete
Are you sure you want to delete zone a-zone (y/[n])? y
```


非大域ゾーンのインストール、停止処理、停止、アンインストール、クローニングについて

この章では、Oracle Solaris オペレーティングシステムへのゾーンのインストールについて説明します。また、仮想プラットフォームおよびアプリケーション環境を管理する2つのプロセス `zoneadm` および `zschd` についても説明します。ゾーンの停止、リブート、クローニング、およびアンインストールに関する情報も提供します。

この章の内容は次のとおりです。

- 37 ページの「ゾーンのインストールと管理の概要」
- 38 ページの「ゾーンの構築」
- 41 ページの「`zoneadm` デーモン」
- 42 ページの「`zschd` ゾーンスケジューラ」
- 42 ページの「ゾーンアプリケーション環境」
- 43 ページの「ゾーンの停止処理、停止、リブート、およびアンインストールについて」
- 45 ページの「非大域ゾーンのクローニングについて」

非大域ゾーンのクローン、非大域ゾーンのインストールとブート、および非大域ゾーンの停止やアンインストールの手順については、第3章「非大域ゾーンのインストール、ブート、停止処理、停止、アンインストール、およびクローニング」を参照してください。

`solaris10` ブランドゾーンのインストールについては、『Oracle Solaris 10 ゾーンの実成と使用』の第5章、「`solaris10` ブランドゾーンのインストール」を参照してください。

ゾーンのインストールと管理の概要

`zoneadm` コマンド (`zoneadm(1M)` のマニュアルページに記載されています) は、非大域ゾーンをインストールおよび管理するために使用される主なツールです。`zoneadm` コマンドを使用する操作は、大域ゾーンから実行します。役割に基づくアクセス

ス制御 (RBAC) を使用中の場合、別のゾーンのコピーを作成するサブコマンドには、`solaris.zone.clonefrom/source_zone` の承認が必要です。

`zoneadm` コマンドを使用して、次のタスクを実行します。

- ゾーンを検証します
- ゾーンをインストールします
- ゾーンを接続します
- インストールされているゾーンの状態を「不完全」に変更します。
- ゾーンをブートします。これは、通常の Oracle Solaris システムのブートに似ています。
- 稼働中のゾーンに関する情報を表示します
- ゾーンを停止処理します
- ゾーンを停止します
- ゾーンをリブートします
- ゾーンをアンインストールします
- 同じシステム上で、ゾーンを別の場所へ再配置します
- 同一システムの既存ゾーンの構成に基づいて、新しいゾーンをプロビジョニングします
- ゾーンを移行します。 `zonecfg` コマンドとともに使用します

ゾーンのインストールおよび確認手順については、[第3章「非大域ゾーンのインストール、ブート、停止処理、停止、アンインストール、およびクローニング」](#) および [`zoneadm\(1M\)` のマニュアルページ](#) を参照してください。 `zoneadm list` コマンドでサポートされるオプションについては、[`zoneadm\(1M\)` のマニュアルページ](#) も参照してください。ゾーンの構成手順については、[第1章「非大域ゾーンの計画および構成方法」](#) および [`zonecfg\(1M\)` のマニュアルページ](#) を参照してください。ゾーンの状態については、『[Oracle Solaris ゾーン](#)の紹介』の「[非大域ゾーンの状態モデル](#)」に記載されています。

ゾーンの Oracle Solaris 監査レコードを生成する場合は、非大域ゾーンをインストールする前に [146 ページ](#) の「[ゾーン内での Oracle Solaris 監査の使用](#)」を参照してください。

ゾーンの構築

このセクションの内容は、既存のゾーンのクローニングにではなく、初期のゾーン構築に適用されます。

ゾーンは、`zoneadm install -m` コマンドに渡されるマニフェストによって指定されたパッケージを使用してインストールされます。マニフェストが指定されない場合、デフォルトのマニフェストは `pkg:/group/system/solaris-small-server` を使

用します。新しいゾーンは、デフォルトの `solaris` 構成とログ (SMF リポジトリの `/etc`、`/var`) を保持し、これらは、`zoneadm install -s` に渡されたプロファイル、およびすべての `zonecfg add net` エントリに指定されたネットワーク接続情報によって変更されるだけです。

システムリポジトリ、ゾーンの構成されたパブリッシャー、および大域ゾーンとの同期が維持されたパッケージについては、[第8章「ゾーンがインストールされている Oracle Solaris システムでの自動インストールおよびパッケージ」](#)を参照してください。

ゾーンのルートファイルシステムに必要とされるファイルは、システムによりゾーンのルートパス内にインストールされます。

正常にインストールされたゾーンは、ブートと初期ログインの準備が整っています。

ゾーンのインストール時に、次の場所からデータが参照またはコピーされます。

- インストールされていないパッケージ
- CD および DVD 内のデータ
- ネットワークインストールイメージ

また、大域ゾーンに存在可能な、次のタイプの情報は、インストール中のゾーンにはコピーされません。

- `/etc/passwd` ファイル内の新規または変更されたユーザー
- `/etc/group` ファイル内の新規または変更されたグループ
- DHCP アドレスの割り当てなどのネットワークサービスの構成
- `sendmail` などのネットワークサービスのカスタマイズ
- ネームサービスなどのネットワークサービスの構成
- 新規または変更された `crontab`、プリンタ、およびメールファイル
- システムログ、メッセージ、およびアカウンティングファイル

Oracle Solaris 監査を使用する場合、ファイルへの変更が必要になる場合があります。詳細は、[146 ページの「ゾーン内での Oracle Solaris 監査の使用」](#)を参照してください。

ゾーンの状態がインストール済みから準備完了に移行する際、構成ファイルで指定されたリソースセットが追加されます。システムにより、一意のゾーン ID が割り当てられます。ファイルシステムがマウントされ、ネットワークインタフェースが設定され、デバイスが構成されます。準備完了状態に移行すると、仮想プラットフォームでユーザープロセスを開始する準備が整います。準備完了状態では、仮想プラットフォームを管理するため、`zsched` および `zoneadmd` プロセスが開始されます。

- `sched` に類似したシステムスケジューリングプロセスである `zsched` が、ゾーンに関連付けられたカーネルリソースの追跡に使用されます。
- `zoneadmd` は、ゾーン管理デーモンです。

準備完了状態のゾーンには、そのゾーン内で実行中のユーザープロセスは存在しません。ゾーンを実行状態にするには、そのゾーン内で少なくとも1つのプロセスが実行中である必要があります。詳細は、[init\(1M\)](#) のマニュアルページを参照してください。

ゾーンのインストール方法

solaris ブランドのインストーラは、次の方法を使用してゾーンのインストールをサポートします。

- パブリッシャーの起点です。
非大域ゾーンをインストールするには、solaris パブリッシャーの起点として設定するリポジトリが、大域ゾーンがインストールされているものと同じソフトウェアを最低限含んでいる必要があります。
- Oracle Solaris リリースが稼働しているインストール済みシステムまたは solaris 非大域ゾーンの統合アーカイブファイルイメージ。
- ゾーンの BE (`zoneadm install -z zbe` を使用)。システムは、必要に応じてパッケージ更新を実行します。

次の表に、インストーラオプションを示します。例のコマンド行については、[51 ページの「構成済みのゾーンをインストールする方法」](#)を参照してください。

オプション	説明
<code>-m manifest</code>	AI マニフェストは、ゾーンのインストール方法を定義する XML ファイルです。ファイルの引数は絶対パスで指定する必要があります。
<code>-c profile dir</code>	構成時に適用する、プロファイルまたはプロファイルのディレクトリを指定します。ファイルの引数は絶対パスで指定する必要があります。プロファイルが適用されると、構成手順が非対話的に実行されます。プロファイルを指定しないと、システムの構成に対話型システム構成ツールが使用されます。すべてのプロファイルは、.xml ファイル拡張子を持つ必要があります。-c にディレクトリオプションを指定すると、そのディレクトリ内のすべてが有効で、正しく形成された構成プロファイルである必要があります。
<code>-a archive</code>	非大域ゾーンをインストールするために使用されるアーカイブへのパス。アーカイブは <code>gzip</code> または <code>bzip</code> を使用して圧縮できます。-d オプションと -a オプションは両立しません。 -a archive オプションを使用すると、必要に応じてパッケージの更新が行われます。必要な場合

オプション	説明
-d <i>path</i>	<p>は、<code>zoneadm attach</code> サブコマンドを使用して、ゾーンをその元のホストに再接続できます。</p> <p>インストール済みシステムまたは非大域ゾーンのルートディレクトリのパス。必要に応じて、パッケージの更新が行われます。<i>path</i> にハイフン (-) を指定すると、<code>zonepath</code> にシステムイメージがすでに配置されているとみなされます。-d オプションと -a オプションは両立しません。</p>
-p	<p>ゾーンのインストール後にシステム ID を保持します。-p オプションと -u オプションは両立しません。</p>
-s	<p>サイレントインストールします。-s オプションと -v オプションは両立しません。</p>
-u	<p>インストール後にゾーンを構成解除し、ゾーンブート時に新しい構成を求めるプロンプトを表示します。-p オプションと -u オプションは両立しません。</p>
-U	<p>必要に応じて、すべてのパッケージを最新のバージョンに更新して、大域ゾーンにインストールされているパッケージとの互換性を持たせます。</p>
-v	<p>インストールプロセスの詳細情報を出力します。-s オプションと -v オプションは両立しません。</p>
-x	<p><code>force-zpool-import</code> を -x オプションとともに使用すると、使用中と見られる <code>zpool</code> が強制的にインポートされます。</p> <p>ストレージオブジェクトに既存のパーティション、<code>zpool</code>、または UFS ファイルシステムが含まれる場合は、<code>install</code> が失敗し、エラーメッセージが表示されます。<code>zoneadm install</code> に対して -x オプションを使用すると、インストールが続行され、既存のデータが上書きされます。このオプションは、<code>zpool create -f</code> コマンドに似ています。</p> <p><code>force-zpool-create-all</code> を -x オプションとともに使用すると、すべての <code>zpool</code> リソースが強制的に作成されます。このオプションを特定の <code>zpool</code> または <code>zpool</code> のセットに制限するには、<code>force-zpool-create=zpoolname</code> を使用します。使用方法については、<code>zoneadm(1M)</code> のマニュアルページを参照してください。</p>

zoneadmd デーモン

ゾーン管理デーモン `zoneadmd` は、ゾーンの仮想プラットフォーム管理用の主要なプロセスです。このデーモンは、ゾーンのブートおよび停止処理の管理も担当します。システム上のアクティブな (準備完了、稼働中、または停止処理中の) ゾーンごとに、1 つの `zoneadmd` プロセスが存在します。

zoneadmd デーモンは、ゾーン構成での指定に従ってゾーンを設定します。このプロセスには、次の処理が含まれます。

- ゾーン ID を割り当てて、zsched システムプロセスを開始する
- ゾーン規模のリソース制御を設定する
- ゾーン構成の指定に従ってゾーンのデバイスを準備する
- ネットワークインタフェースを設定する
- ループバックおよび従来のファイルシステムをマウントする
- ゾーンコンソールデバイスをインスタンス化および初期化する

zoneadmd デーモンは、まだ実行されていなければ zoneadm によって自動的に起動されます。そのため、このデーモンが何らかの理由で実行されていない場合は、ゾーンを管理するための zoneadm のいずれかの呼び出しによって zoneadmd が再起動されず。

zoneadmd デーモンのマニュアルページは、zoneadmd(1M) です。

zsched ゾーンスケジューラ

アクティブなゾーンとは、準備完了状態、稼働状態、または停止処理状態のゾーンのことです。すべてのアクティブなゾーンには、カーネルプロセス zsched が関連付けられています。ゾーンのために処理を実行するカーネルスレッドは、zsched により所有されています。zsched プロセスにより、ゾーンサブシステムがカーネルスレッドをゾーンごとに追跡することが可能になります。

ゾーンアプリケーション環境

ゾーンアプリケーション環境の作成には、zoneadm コマンドが使用されます。

ゾーンの内部構成は、sysconfig インタフェースを使用して指定します。内部構成では、使用するネームサービス、デフォルトのロケールおよびタイムゾーン、ゾーンのルートパスワード、およびほかのアプリケーション環境特性を指定します。sysconfig インタフェースについては、『[Oracle Solaris 11.3 システムのインストール](#)』の第 6 章、『[Oracle Solaris インスタンスの構成解除または再構成](#)』および [sysconfig\(1M\)](#) のマニュアルページに記載されています。ゾーンのデフォルトロケールおよびタイムゾーンは、大域設定には関係なく構成できます。

ゾーンの停止処理、停止、リブート、およびアンインストールについて

このセクションでは、ゾーンの停止、リブート、アンインストール、およびクローニングの概要について説明します。

ゾーンの停止処理

`zoneadm shutdown c` コマンドは、完全にゾーンを停止処理するために使用します。このアクションは、ゾーン内で `/usr/sbin/init 0` を実行するのと同様です。-r オプションも指定すると、ゾーンはリブートされます。サポートされるブートオプションについては、[44 ページの「ゾーンのブート引数」](#)を参照してください。

`svc:/system/zones` サービスは、大域ゾーンの停止処理時に、`zoneadm shutdown` を使用してゾーンを完全に停止処理します。

`shutdown` サブコマンドはゾーンが正常に停止処理されるまで待機します。このアクションが適切な時間内に完了しない場合は、`zoneadm halt` を使用すると、そのゾーンを強制的に停止できます。詳細は、[58 ページの「ゾーンの停止方法」](#)を参照してください。

ゾーンの停止

`zoneadm halt` コマンドを使用して、ゾーン内で実行されているすべてのプロセスを終了し、仮想プラットフォームを削除します。これにより、ゾーンはインストール済みの状態に戻されます。すべてのプロセスが終了し、デバイスが構成解除され、ネットワークインタフェースが破棄され、ファイルシステムがアンマウントされ、カーネルデータ構造が破棄されます。

`halt` コマンドにより、ゾーン内部の停止処理スクリプトが実行されることはありません。ゾーンを停止処理するには、[43 ページの「ゾーンの停止処理」](#)を参照してください。別の方法として、ゾーンにログインして停止処理を実行することもできます。[82 ページの「zlogin を使用してゾーンを停止処理する方法」](#)を参照してください。

停止操作に失敗する場合は、[186 ページの「ゾーンが停止しない」](#)を参照してください。

ゾーンのリブート

zoneadm reboot コマンドを使用してゾーンをリブートします。ゾーンのリブート時に、ゾーン ID が変更されます。

ゾーンのブート引数

zoneadm boot および reboot サブコマンドでは、次のブート引数を使用できます。

- `-i altinit`
- `-m smf_options`
- `-s`

次の定義が適用されます。

`-i altinit` 最初のプロセスとなる代替実行可能ファイルを選択します。`altinit` は実行可能ファイルへの有効なパスでなければなりません。デフォルトの最初のプロセスについては、[init\(1M\)](#) のマニュアルページに記載されています。

`-m smf_options` SMF のブート動作を制御します。復元オプションとメッセージオプションという、2 種類のオプションがあります。メッセージオプションは、ブート中に表示されるメッセージの種類と数を決定します。サービスオプションは、システムのブートに使用されるサービスを決定します。

復元オプションは次のとおりです。

`デバッグ` 標準のサービス別出力と、ログに記録されるすべての `svc.startd` メッセージを出力します。

`milestone=milestone` 指定されたマイルストーンで定義されているサブグラフにブートします。有効なマイルストーンは、`none`、`single-user`、`multi-user`、`multi-user-server`、および `all` です。

メッセージオプションは次のとおりです。

`quiet` 標準のサービス別出力と、管理者の介入を必要とするエラーメッセージを出力します。

`verbose` 標準のサービス別出力と、詳細情報を提供するメッセージを出力します。

-s マイルストーン `svc:/milestone/single-user:default` に対してのみブートします。このマイルストーンは、`init` のレベル `s` と同等です。

使用例については、56 ページの「ゾーンのブート方法」および 57 ページの「ゾーンをシングルユーザーモードでブートする方法」を参照してください。

Oracle Solaris サービス管理機能 (SMF) および `init` については、『Oracle Solaris 11.3 でのシステムサービスの管理』、`svc.startd(1M)`、および `init(1M)` を参照してください。

ゾーンの autoboot 設定

大域ゾーンのブート時にゾーンを自動的にブートするには、ゾーンの構成内の `autoboot` リソースプロパティを `true` に設定します。デフォルトの設定は `false` です。

ゾーンを自動的にブートするには、ゾーンサービス `svc:/system/zones:default` も有効になっている必要があります。このサービスは、デフォルトで有効になっています。

`pkg update` の実行中の `autoboot` の設定については、109 ページの「ゾーンのパッケージングの概要」を参照してください。

ゾーンのアンインストール

ゾーンのルートファイルシステム内のすべてのファイルをアンインストールする場合には、`zoneadm uninstall` コマンドを使用します。`-F (force)` オプションを合わせて指定しない限り、処理を続行する前に、コマンドプロンプトにより実行の確認が求められます。実行した操作を元に戻すことはできないため、`uninstall` コマンドは慎重に使用してください。

非大域ゾーンのクローニングについて

クローンを使用すると、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コピーして、新しいゾーンを同一のシステム上に迅速にプロビジョニングできます。少なくとも、複数のゾーンで同一であってはならないコンポーネントに対しては、プ

ロパティールリソースをリセットする必要があります。したがって、`zonepath` は常に変更する必要があります。さらに、共有 IP ゾーンの場合は、各 `net` リソースの IP アドレスが異なっている必要があります。排他的 IP ゾーンの場合は、各 `net` リソースの `physical` プロパティが異なっている必要があります。通常、アプリケーション固有の構成はクローンで再構成する必要があります。たとえば、ゾーン内にデータベースインスタンスがあり、そのゾーンをクローニングする場合は、クローン内のデータベースインスタンスを、別のインスタンスとして自身を認識するように再構成しなければならない可能性があります。

- ゾーンのクローニングは、ゾーンのインストールほど時間がかかりません。
- 新規ゾーンには、パッケージの追加やファイルの変更など、ソースゾーンをカスタマイズする過程で加えられた変更がすべて含まれます。

次の方法のいずれかを使用すると、ゾーンをクローンできます。

- `zoneadm clone` コマンドを使用してゾーンをクローニングします。この方法は、少数のゾーンをクローニングする場合に推奨されます。

クローニング元の `zonepath` とクローニング先の `zonepath` が両方とも ZFS 上にあり、同じプールに含まれる場合、`zoneadm clone` コマンドは自動的に ZFS を使用してゾーンをクローンします。ZFS クローンを使用する場合、データが変更されるまでデータは実際にはコピーされません。したがって、最初のクローンにかかる時間はごくわずかです。`zoneadm` コマンドは、ソース `zonepath` の ZFS スナップショットを取得して、ターゲット `zonepath` を設定します。ZFS クローンの名前には、宛先ゾーンの `zonepath` が使用されます。

注記 - ZFS クローンを使用してソースを複製できる場合でも、ZFS クローンは行わず ZFS の `zonepath` をコピーするように指定することができます。

詳細は、61 ページの「[同一システム上での非大域ゾーンのクローニング](#)」を参照してください。

- 統合アーカイブファイルを使用したゾーンのクローニング。大規模な配備で複数のゾーンをクローンする必要があるときは、この方法が推奨されます。この方法でのゾーンのクローニングには、次の手順が必要です。
 1. 統合アーカイブファイルの作成。統合アーカイブファイルには、すべてのゾーンを含めることも、選択したゾーンを含めることもできます。
 2. `zonecfg` および `zoneadm` コマンドを使用した、1 つまたは複数の新しいゾーンの構成およびインストール。参照ソースとしてアーカイブを使用して新しいゾーンを作成すると、新しいゾーンは元のシステム構成を模倣します。

詳細は、『[Oracle Solaris 11.3 でのシステム復旧とクローン](#)』の第 1 章、「[Oracle Solaris のシステム回復とクローニング \(概要\)](#)」を参照してください。

ゴールデンゾーンを作成する

定期的にクローンする特定のゾーンがある場合は、ゾーンのクローンを作成し、それを使用して将来のクローンを作成できます。このゾーンをゴールデンゾーンと呼びます。

ゾーン `z1` からゴールデンゾーンを作成するには:

1. `# zonecfg -z z1-golden 'create -t z1; set zonepath=/zones/z1-golden'`
2. `# zoneadm -z z1 shutdown`
3. `# zoneadm -z z1-golden clone z1`
4. `# zoneadm -z z1 boot`

ゴールデンゾーン `z1-golden` のコピーを作成するには:

1. `# zonecfg -z z2 'create -t z1-golden; set zonepath=/zones/z2'`
2. `# zoneadm -z z2 clone z1-golden`

クローニング中にソースゾーンをシャットダウンする必要はありません。これまでにブートされたことがない `z1-golden` があるため、作成するために使用されたスナップショットからはほとんど分岐していません。そのため、`z1-golden` のコストはディスク容量 1 メガバイト未満であると考えられます。新しいブート環境が作成されると、更新されたパッケージで使用されるディスク容量のコストのみで、`z1-golden` は新しいブート環境を取得できます。

非大域ゾーンのインストール、ブート、停止処理、停止、アンインストール、およびクローニング

この章では、非大域ゾーンのインストールおよびブート方法について説明します。クローニングを使って同一のシステムにゾーンをインストールする方法についても説明します。ゾーンの停止、リブート、アンインストールなどのインストールに関連するほかのタスクについて説明します。既存の非大域ゾーンを同じマシンの新しい場所に移動します。また、既存の非大域ゾーンを同じマシン上の新しい場所に移動し、システムからゾーンを完全に削除する手順も含まれています。

ゾーンのインストールおよび関連する操作に関する一般的な情報については、[第2章「非大域ゾーンのインストール、停止処理、停止、アンインストール、クローニングについて」](#)を参照してください。

solaris10 ブランドゾーンのインストールおよびクローニングについては、『[Oracle Solaris 10 ゾーンの作成と使用](#)』の第5章、「[solaris10 ブランドゾーンのインストール](#)」を参照してください。

ゾーンのインストール

タスク	説明	手順
(オプション) ゾーンをインストールする前に、構成済みのゾーンを検証します。	ゾーンがインストール要件を満たしていることを確認します。この手順を省略した場合、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。	50 ページの「(オプション)インストール前に構成済みのゾーンを検証する方法」
構成済みのゾーンをインストールします。	構成済みの状態にあるゾーンをインストールします。	51 ページの「構成済みのゾーンをインストールする方法」
ゾーンの汎用一意識別子 (UUID) を取得します。	ゾーンのインストール時に割り当てられるこの個別の識別子は、ゾーンを識別するための代替手段になります。	53 ページの「インストールされた非大域ゾーンの UUID を取得する方法」

タスク	説明	手順
(オプション) インストール済みのゾーンを準備完了状態に移行します。	ゾーンをすぐにブートして使用する場合、この手順は省略できます。	55 ページの「(オプション) インストール済みのゾーンを準備完了状態に移行する方法」
ゾーンをブートします。	ゾーンをブートすると、ゾーンが稼働状態になります。ゾーンは、準備完了状態またはインストール済み状態からブートできます。	56 ページの「ゾーンのブート方法」
ゾーンをシングルユーザーモードでブートします。	マイルストーン <code>svc:/milestone/single-user:default</code> に対してのみブートします。このマイルストーンは、 <code>init</code> のレベル <code>s</code> と同等です。 init(1M) および svc.startd(1M) のマニュアルページを参照してください。	57 ページの「ゾーンをシングルユーザーモードでブートする方法」

ゾーンのインストールとブート

`zoneadm(1M)` のマニュアルページで説明されている `zoneadm` コマンドを使用して、非大域ゾーンのインストールタスクを実行します。ゾーンのインストールを実行するには、大域管理者または適切な承認を持つユーザーである必要があります。この章に示す例では、[22 ページの「ゾーンを構成、検証、および確定する」](#)で使用したゾーン名およびゾーンパスを使用します。

▼ (オプション) インストール前に構成済みのゾーンを検証する方法

ゾーンをインストールする前に検証できます。実行する検査の 1 つは、十分なディスクサイズがあるかどうかの検査です。この手順を省略した場合、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。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. **管理者になります。**
詳細は、『[Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護](#)』の「[割り当てられている管理権利の使用](#)」を参照してください。
2. **-z オプションをゾーン名および `verify` サブコマンドとともに使用して、`my-zone` という名前の構成済みゾーンを検証します。**

```
global# zoneadm -z my-zone verify
```

ゾーンパスの検証に関する次のメッセージが表示されます。

```
WARNING: /zones/my-zone does not exist, so it could not be verified.
```

```

When 'zoneadm install' is run, 'install' will try to create
/systems/my-zone, and 'verify' will be tried again,
but the 'verify' may fail if:
the parent directory of /system/zones/my-zone is group- or other-writable
or
/system/zones/my-zone overlaps with any other installed zones
or
/system/zones/my-zone is not a mountpoint for a zfs file system.

```

ただし、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。

共有ストレージ上のゾーンの検証

共有ストレージに構成されているゾーンでは、ゾーンが構成済み状態の場合、`zonecfg verify` によって、構成されているどの `zpool` リソースもシステム上でまだオンラインになっていないことが検証されます。

共有ストレージに構成されているゾーンでは、ゾーンがインストール済み状態の場合は、`zoneadm verify` コマンドによって、`zpool` および `rootzpool` リソースとして構成されたすべての `zpool` がシステム上でオンラインになっていることが検証されます。リソースが使用できない場合は、`verify` が失敗し、失敗した `zpool` に関する情報が表示されます。

▼ 構成済みのゾーンをインストールする方法

この手順を使用して、構成済みの非大域ゾーンをインストールします。インストールオプションについては、[40 ページの「ゾーンのインストール方法」](#)を参照してください。

ゾーンは、独自の ZFS データセット上に存在する必要があります。ZFS のみがサポートされます。`zoneadm install` コマンドは、ゾーンのインストール時に `zonepath` に対して ZFS ファイルシステム (データセット) を自動的に作成します。ZFS データセットを作成できない場合、ゾーンのインストールは行われません。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. `zoneadm` コマンドを `install` サブコマンドとともに使用し、`zonepath ZFS` に自動的に ZFS データセットを作成して、構成済みのゾーン `my-zone` をインストールしま

す。ゾーンのパスの親ディレクトリもデータセットである必要があり、データセットでない場合はファイルシステムの作成が失敗することに注意してください。

- ゾーンをインストールします。

```
global# zoneadm -z my-zone install
```

- リポジトリからゾーンをインストールします。

```
global# zoneadm -z my-zone install -m manifest -c [ profile | dir ]
```

- 次のコマンドを実行して、イメージからゾーンをインストールします。

```
global# zoneadm -z my-zone install -a archive -s -u
```

- 次のコマンドを実行して、ディレクトリからゾーンをインストールします。

```
global# zoneadm -z my-zone install -d path -p -v
```

このゾーンに対して ZFS ファイルシステムが作成されたことが表示されます。

ゾーンのルートファイルシステムに必要なファイルおよびディレクトリがゾーンのルートパスにインストールされる際、さまざまなメッセージが表示されます。

3. (オプション) エラーメッセージが表示され、ゾーンのインストールに失敗した場合は、次のように入力してゾーンの状態を取得します。

```
global# zoneadm list -v
# zoneadm list -cvd
ID NAME          STATUS    PATH                                BRAND  IP
0  global         running  /                                    solaris shared
-  my-zone        configured /zones/my-zone                    solaris excl
```

- 状態が構成済みであると表示された場合は、メッセージに示された修正を行い、`zoneadm install` コマンドを再度実行します。
- 状態が不完全であると表示された場合は、最初に次のコマンドを実行します。

```
global# zoneadm -z my-zone uninstall
```

メッセージに示された修正を行い、`zoneadm install` コマンドを再度実行します。

4. (オプション) ストレージオブジェクトに既存のパーティション、`zpool`、または UFS ファイルシステムが含まれる場合は、`install` が失敗し、エラーメッセージが表示されます。

`force` サブコマンドを使用する前に、ソースゾーンをアンインストール済みの状態にする必要があります。

```
zoneadm -z my-zone uninstall
```

その後、`zoneadm install` に対して `-x` オプションを使用して、インストールを続行し、既存のデータを上書きします。

```
-x force-zpool-import
-x force-zpool-create=zpoolname
-x force-zpool-create=zpoolname1,zpoolname2,zpoolname3
-x force-zpool-create-all
```

このオプションは、`zpool create -f` コマンドに似ています。

`-x force-zpool-create=zpoolname` は、1 回または複数回使用できます。

5. インストールが完了したら、`list` サブコマンドに `-i` オプションおよび `-v` オプションを指定してインストール済みのゾーンを一覧表示し、ステータスを確認します。

```
global# zoneadm list -iv
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	my-zone	installed	/zones/my-zone	solaris	excl

注意事項 ゾーンの実インストールが中断または失敗した場合は、ゾーンの状態は不完全なままになります。`uninstall -F` を使用して、ゾーンを構成済みの状態にリセットします。

次の手順 このゾーンは、Oracle Solaris サービス管理機能 (SMF) フレームワークでインストールされました。『Oracle Solaris 11.3 でのシステムサービスの管理』の第 3 章、「サービスの管理」の説明に従って、個々のサービスを有効または無効にできます。

▼ インストールされた非大域ゾーンの UUID を取得する方法

ゾーンの実インストール時に、汎用一意識別子 (UUID) がゾーンに割り当てられます。UUID は、`zoneadm` を `list` サブコマンドおよび `-c -p` オプションとともに使用して取得します。UUID は、5 番目に表示されるフィールドです。

- インストールされたゾーンの UUID を表示します。

```
global# zoneadm list -cp
```

次のような情報が表示されます。

```
0:global:running:/::solaris:shared:-:none:
3:test_zone:running:/system/volatile/zones/test_zone/zonepath:95180a6d-fab2-4363-ee33-81ba6e84a84f:solaris-kz:excl:R:solaris-kz:
-:zone123:installed:/system/zones/zone123:96972ce7-d41d-4fec-ff4b-8f14123e0974:solaris:excl:-:
```

例 1 特定のゾーンの UUID を取得する方法

次のコマンドを使用して、`test_zone` の UUID を取得します。

```
# zoneadm list -cp | grep test_zone | cut -f 5 -d:
```

次のような情報が表示されます。

```
95180a6d-fab2-4363-ee33-81ba6e84a84f
```

例 2 コマンドで `test_zone` UUID を使用する方法

```
global# zoneadm -z test_zone -u 95180a6d-fab2-4363-ee33-81ba6e84a84f list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
3	test_zone	running	-	solaris-kz	excl

`-u uuid-match` と `-z zonename` の両方が存在する場合、最初に UUID に基づいてマッチングが行われます。指定した UUID のゾーンが見つかった場合はそのゾーンが使用され、`-z` パラメータは無視されます。指定した UUID のゾーンが見つからなかった場合、システムはゾーン名で検索を実行します。

UUID について

ゾーンをアンインストールすることも、同名のゾーンを内容を変えて再インストールすることもできます。ゾーンの内容を変更せずにゾーンの名前を変更することも可能です。こうした理由から、UUID はゾーン名よりも信頼性が高くなります。

参照 詳細は、[zoneadm\(1M\)](#) および [libuuid\(3LIB\)](#) を参照してください。

▼ インストールした非大域ゾーンに不完全のマークを付ける方法

システムに加えられた管理上の変更のためにゾーンが使用不可になるか、矛盾が生じた場合、インストールしたゾーンの状態を不完全に変更できます。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. ゾーン `testzone` に不完全のマークを付けます。

```
global# zoneadm -z testzone mark incomplete
```

3. `list` サブコマンドに `-i` オプションと `-v` オプションを使って、ステータスを検証します。

```
global# zoneadm list -iv
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	my-zone	installed	/zones/my-zone	solaris	excl
-	testzone	incomplete	/zones/testzone	solaris	excl

ゾーンへの不完全のマーク付け

-R root オプションを、zoneadm の mark サブコマンドや list サブコマンドとともに使用して、代替ブート環境を指定できます。詳細は、[zoneadm\(1M\)](#) のマニュアルページを参照してください。

注記 - ゾーンへの不完全のマーク付けは、取り消すことができません。不完全のマークが付けられたゾーンに実行可能なのは、ゾーンをアンインストールして、構成済みの状態に戻す操作だけです。[60 ページの「ゾーンをアンインストールする方法」](#)を参照してください。

▼ (オプション) インストール済みのゾーンを準備完了状態に移行する方法

準備完了状態に移行すると、仮想プラットフォームでユーザープロセスを開始する準備が整います。準備完了状態のゾーンには、内部で実行中のユーザープロセスは存在しません。

ゾーンをすぐにブートして使用する場合、この手順は省略できます。ゾーンのブート時に、準備完了状態への移行が自動的に行われます。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. zoneadm コマンドを -z オプション、ゾーン名 (my-zone)、および ready サブコマンドとともに使用することで、そのゾーンを準備完了状態に移行します。

```
global# zoneadm -z my-zone ready
```

3. プロンプトで、zoneadm list コマンドに -v オプションを指定して、ステータスを検証します。

```
global# zoneadm list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	my-zone	ready	/zones/my-zone	solaris	excl

システムにより一意のゾーン ID 1 が割り当てられていることに注目してください。

▼ ゾーンのブート方法

ゾーンを実行状態にするには、そのゾーンをブートします。準備完了状態またはインストール済み状態からブートできます。ブートしたインストール済み状態のゾーンは、準備完了状態から稼働状態に透過的に移行します。稼働状態のゾーンに対してはゾーンへのログインが可能です。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. zoneadm コマンドを -z オプション、ゾーン名 (my-zone)、および boot サブコマンドとともに使用することで、ゾーンをブートします。

```
global# zoneadm -z my-zone boot
```

3. ブートが完了したら、list サブコマンドに -v オプションを指定してステータスを確認します。

```
global# zoneadm list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	my-zone	running	/zones/my-zone	solaris	excl

例 3 ゾーンのブート引数を指定する

-m verbose オプションを使用してゾーンをブートします。

```
global# zoneadm -z my-zone boot -- -m verbose
```

-m verbose ブートオプションを使用してゾーンをリブートします。

```
global# zoneadm -z my-zone reboot -- -m verbose
```

ゾーン管理者が `-m verbose` オプションを使用してゾーン `my-zone` をリブートします。

```
my-zone# reboot -- -m verbose
```

▼ ゾーンをシングルユーザーモードでブートする方法

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. ゾーンをシングルユーザーモードでブートします。

```
global# zoneadm -z my-zone boot -- -s
```

次に進む手順

ゾーンにログインして初期内部構成を実行する方法については、[第4章「非大域ゾーンへのログインについて」](#) および [第5章「非大域ゾーンへのログイン」](#) を参照してください。

ゾーンの停止処理、停止、リブート、およびアンインストール

▼ ゾーンを停止処理する方法

停止処理手順は、ゾーンを完全に停止処理します。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. システムで稼働中のゾーンの一覧を表示します。

```
global# zoneadm list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	my-zone	running	/zones/my-zone	solaris	excl

3. **zoneadm** コマンドを **-z** オプション、ゾーン名 (**my-zone** など)、および **shutdown** サブコマンドとともに使用することで、指定されたゾーンを停止処理します。

```
global# zoneadm -z my-zone shutdown
```

4. また、**-r** オプションを指定してゾーンをリポートします。

```
global# zoneadm -z my-zone shutdown -r boot_options
```

例3「ゾーンのブート引数を指定する」を参照してください。

5. システム上で稼働するゾーンをリスト表示して、ゾーンが停止処理されたことを確認します。

```
global# zoneadm list -v
```

▼ ゾーンの停止方法

この停止手順を使用して、ゾーンのアプリケーション環境と仮想プラットフォームの両方を削除します。ゾーンの完全な停止処理を行う方法については、[82 ページの「zlogin を使用してゾーンを停止処理する方法」](#)を参照してください。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. システムで稼働中のゾーンの一覧を表示します。

```
global# zoneadm list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	my-zone	running	/zones/my-zone	solaris	excl

3. **zoneadm** コマンドを **-z** オプション、ゾーン名 (**my-zone** など)、および **halt** サブコマンドとともに使用することで、指定されたゾーンを停止します。

```
global# zoneadm -z my-zone halt
```

4. システム内のゾーンの一覧を再度表示して、**my-zone** が停止していることを確認します。

```
global# zoneadm list -iv
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	my-zone	installed	/zones/my-zone	solaris	excl

5. ゾーンを再び起動する場合は、次のコマンドをブートします。

```
global# zoneadm -z my-zone boot
```

注意事項 ゾーンが正しく停止しない場合は、[186 ページの「ゾーンが停止しない」](#) でトラブルシューティングのヒントを参照してください。

▼ ゾーンをリポートする方法

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。[57 ページの「ゾーンを停止処理する方法」](#) も参照してください。

1. 管理者になります。

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

2. システムで稼働中のゾーンの一覧を表示します。

```
global# zoneadm list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	my-zone	running	/zones/my-zone	solaris	excl

3. **zoneadm** コマンドを **-z reboot** オプションとともに使用することで、ゾーン **my-zone** をリポートします。

```
global# zoneadm -z my-zone reboot
```

4. システム内のゾーンの一覧を再度表示して、**my-zone** がリポートしたことを確認します。

```
global# zoneadm list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
2	my-zone	running	/zones/my-zone	solaris	excl

ヒント - my-zone のゾーン ID が変更されていることに注目してください。通常、リブートするとゾーン ID は変更されます。

▼ zoneadm コマンドを使用してゾーンの名前を変更する方法

ゾーンの名前を変更するには、zoneadm コマンドと rename サブコマンドを使用します。

1. 管理者になります。

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

2. ゾーンの名前を変更します。

```
zoneadm -z old_name rename new_name
```

▼ ゾーンをアンインストールする方法



注意 - この手順は、注意して使用してください。ゾーンのルートファイルシステム内のファイルすべてを削除した後で、操作を元に戻すことはできません。

ゾーンは稼働状態であってはいけません。uninstall 操作は、稼働中のゾーンに対しては無効です。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. システム内のゾーンの一覧を表示します。

```
global# zoneadm list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	my-zone	installed	/zones/my-zone	solaris	excl

3. **zoneadm** コマンドを **-z uninstall** オプションとともに使用することで、ゾーン **my-zone** を削除します。

-F オプションを使用すると、処理を強制的に実行できます。このオプションが指定されていない場合、システムにより確認を求めるメッセージが表示されます。

```
global# zoneadm -z my-zone uninstall -F
```

zonepath として独自の ZFS ファイルシステムを保持しているゾーンをアンインストールすると、その ZFS ファイルシステムは破棄されます。

4. システム内のゾーンの一覧を再度表示して、**my-zone** が一覧に含まれていないことを確認します。

```
global# zoneadm list -iv
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared

注意事項 ゾーンをアンインストールが中断した場合、ゾーンの状態は不完全なままになります。zoneadm uninstall コマンドを使用して、ゾーンを構成済みの状態にリセットしてください。

zonepath が削除されない場合、このゾーンが別のブート環境内にインストールされていることを示している可能性があります。特定の zonepath を保持するインストール済みゾーンを持つブート環境が存在するとき、zonepath やその zonepath データセット内に存在するさまざまなデータセットは削除されません。ブート環境に関する詳細は、[beadm\(1M\)](#) を参照してください。

実行した操作を元に戻すことはできないため、uninstall コマンドは慎重に使用してください。

同一システム上での非大域ゾーンのクローニング

ソース zonepath からターゲット zonepath にデータをコピーすることによってシステム上に新しいゾーンをプロビジョニングするには、クローニングを使用します。

クローニング元の zonepath とクローニング先の zonepath が両方とも ZFS 上にあり、同じプールに含まれる場合、zoneadm clone コマンドは自動的に ZFS を使用して

ゾーンをクローンします。ただし、ZFS の `zonepath` のコピーは行い、ZFS のクローンは行わないように指定することもできます。

▼ ゾーンをクローンする方法

新規ゾーンをインストールする前に、そのゾーンを構成する必要があります。 `zoneadm create` サブコマンドに渡されるパラメータは、クローンするゾーンの名前です。このソースゾーンを停止する必要があります。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. クローンされるソースゾーン (この手順では `my-zone`) を停止します。

```
global# zoneadm -z my-zone halt
```

3. ソースゾーン `my-zone` の構成をファイル (たとえば、`master`) にエクスポートすることにより、新規ゾーンの構成を開始します。

```
global# zonecfg -z my-zone export -f /zones/master
```

注記 - 既存の構成を変更する代わりに、22 ページの「ゾーンの構成方法」で説明されている手順を使って、新規ゾーン構成を作成することもできます。この方法を使用する場合は、ゾーンを作成したあとで手順 6 に進みます。

4. `master` ファイルを編集します。複数のゾーンで同一であってはならないコンポーネントに対して、異なるプロパティとリソースを設定します。たとえば、新しい `zonepath` を設定する必要があります。共有 IP ゾーンの場合は、各 `net` リソースの IP アドレスを変更する必要があります。排他的 IP ゾーンの場合は、各 `net` リソースの `physical` プロパティを変更する必要があります。

5. `master` ファイル内のコマンドを使って、新規ゾーン `zone1` を作成します。

```
global# zonecfg -z zone1 -f /zones/master
```

6. `my-zone` をクローニングして、新規ゾーン `zone1` をインストールします。

```
global# zoneadm -z zone1 clone my-zone
```

システムには次のように表示されます。

```
Cloning zonepath /zones/my-zone...
```

7. (オプション) ストレージオブジェクトに既存のパーティション、zpool、または UFS ファイルシステムが含まれる場合は、clone が失敗し、エラーメッセージが表示されます。

操作を続行し、既存のデータを上書きするには、zoneadm clone に対して適切な -x オプションを使用します。force サブコマンドを使用する前に、ソースゾーンをアンインストールする必要があります。

```
-x force-zpool-import
-x force-zpool-create=zpoolname
-x force-zpool-create=zpoolname1, zpoolname2, zpoolname3
-x force-zpool-create-all
```

このオプションは、zpool create -f コマンドに似ています。

-x force-zpool-create=zpoolname オプションは、複数回使用できます。

-x force オプションを使用する前に、ソースゾーンを停止する必要があります。

8. システム内のゾーンの一覧を表示します。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	my-zone	installed	/zones/my-zone	solaris	excl
-	zone1	installed	/zones/zone1	solaris	excl

- 例 4 システム構成プロファイルのクローンされたゾーンへの適用

構成プロファイルを含めるには、次のように入力します。

```
# zoneadm -z zone1 clone -c /path/config.xml my-zone
```

構成ファイルの絶対パスを指定する必要があります。

非大域ゾーンの移動

zonepath を変更してゾーンを同じシステムの新しい場所に移動する場合に、ここで説明する手順を使用します。ゾーンは、停止する必要があります。『Oracle Solaris ゾーン構成リソース』の「リソースタイプとプロパティ」に記載されている通常の zonepath の基準が適用されます。

この情報は、solaris10 ブランドゾーンの移動にも適用されます。solaris10 ブランドゾーンについては、『Oracle Solaris 10 ゾーンの実成と使用』を参照してください。

注記 - ほかの BE 内に存在しているゾーンを移動することはできません。最初にそれらの BE を削除でき、またそのゾーンをクローニングすることによって新しいパスに新しいゾーンを作成することもできます。

注記 - rootzpool リソースを含む共有ストレージ上のゾーンをシステム上の別の場所に移動することはできません。zonepath の名前変更はサポートされています。

▼ 共有ストレージ上に存在しないゾーンを移動する方法

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. 移動するゾーン (この手順では db-zone) を停止します。

```
global# zoneadm -z db-zone halt
```

3. zoneadm コマンドを move サブコマンドとともに使用して、ゾーンを新規の zonepath である /zones/db-zone に移動します。

```
global# zoneadm -z db-zone move /zones/db-zone
```

4. パスを検証します。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	my-zone	installed	/zones/my-zone	solaris	excl
-	db-zone	installed	/zones/db-zone	solaris	excl

システムから非大域ゾーンを削除する

このセクションでは、システムからゾーンを完全に削除する手順を説明します。

▼ 非大域ゾーンを削除する方法

1. 次のいずれかの方法を使用して、ゾーンの my-zone を停止処理します。zoneadm shutdown の方法をお勧めします。

■ zoneadm を使用:

```
global# zoneadm -z my-zone shutdown  
my-zone
```

■ zlogin を使用:

```
global# zlogin my-zone shutdown
my-zone
```

2. **my-zone のルートファイルシステムを削除します。**

```
global# zoneadm -z my-zone uninstall -F
```

操作を強制実行する -F オプションは通常不要です。

3. **my-zone の構成を削除します。**

```
global# zonecfg -z my-zone delete -F
```

操作を強制実行する -F オプションは通常不要です。

4. **システム内のゾーンの一覧を表示し、my-zone が一覧に含まれていないことを確認します。**

```
global# zoneadm list -iv
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared

非大域ゾーンへのログインについて

この章では、大域ゾーンからゾーンへのログインについて説明します。

この章の内容は次のとおりです。

- 67 ページの「zlogin コマンド」
- 68 ページの「ゾーンの内部構成」
- 74 ページの「非大域ゾーンへのログイン方法」
- 75 ページの「対話型モードと非対話型モード」
- 75 ページの「フェイルセーフモード」
- 75 ページの「リモートログイン」

手順および使用法については、[第5章「非大域ゾーンへのログイン」](#)を参照してください。使用できるオプションの一覧については、[zlogin\(1\)](#)のマニュアルページを参照してください。

zlogin コマンド

役割に基づくアクセス制御 (RBAC) を使用中の場合、ゾーンコンソールへのアクセスには、`solaris.zone.manage/zonename` の承認が必要です。スラッシュ文字 (/) が前に付いた特定の `zonename` の接尾辞はオプションです。省略すると、承認はすべてのゾーンと一致します。

-c オプションを使用してゾーンコンソールに接続しない限り、zlogin を使用してゾーンにログインすると、新しいタスクが開始されます。1つのタスクを2つのゾーンで実行することはできません。

zlogin コマンドを使用して、大域ゾーンから稼働状態または準備完了状態にある任意のゾーンにログインします。

注記 - 稼働状態にないゾーンへのログインに使用できるのは、zlogin コマンドと -c オプションの組み合わせだけです。

81 ページの「非対話型モードを使用してゾーンにアクセスする方法」に記述されているとおり、ゾーン内での実行を指示するコマンドを指定することで、`zlogin` コマンドを非対話型モードで使用できます。ただし、このコマンドおよびこのコマンドの処理対象となるファイルは、いずれも NFS 上に存在してはなりません。開かれているファイルのいずれか、またはそのアドレス空間のいずれかの部分が NFS 上に存在する場合、コマンドは失敗します。アドレス空間には、コマンドの実行可能ファイル自体またはリンクされたライブラリが含まれます。

`zlogin` コマンドは、大域管理者または大域ゾーン内で操作している、適切な承認を持つユーザーだけによって使用できます。詳細は、`zlogin(1)` のマニュアルページを参照してください。

ゾーンの内部構成

システム構成データは、単一のプロファイル `sc_profile.xml`、または SMF プロファイルの `profiles` ディレクトリとして存在できます。単一のファイルまたはディレクトリの両方が、ゾーンのインストール時に自動化されたインストーラに渡される、ゾーンのシステム構成データについて記述します。ゾーンのインストール時に `sc_profile.xml` ファイルも `profiles` ディレクトリも提供されない場合、`sysconfig` 対話型ツールは、コンソールの `zlogin` コマンドが最初に使用されたときに、このデータについて管理者に確認します。

リポジトリからのインストール後に適用するプロファイルまたはプロファイルのディレクトリを指定するには、`zoneadm attach` コマンドで `-z ZBE` を使用します。指定された既存のゾーンブート環境が接続されます。指定されたゾーンブート環境が異なる大域ゾーンに関連付けられている場合は、指定された `ZBE` のクローンが作成され、`ZBE` のクローンが接続されます。すべてのプロファイルは、`.xml` 拡張子を持つ必要があります。

このリリースは、SMF を使用して構成情報を一元管理します。

Oracle Solaris インスタンスは、インストール中に作成されて構成されます。Oracle Solaris インスタンスは、大域ゾーン内または非大域ゾーン内のいずれかのブート環境として定義されます。`sysconfig` ユーティリティを使用して、Oracle Solaris インスタンスに対して構成タスクを実行でき、また Oracle Solaris インスタンスを構成解除して、このインスタンスを再構成することもできます。`sysconfig` コマンドを使用して、SMF プロファイルを作成できます。

大域ゾーンまたは非大域ゾーン内での Oracle Solaris インスタンスのインストール後または作成後にシステム構成が必要な場合は、システム構成が自動的に実行されます。システム ID を保持するための `-p` オプションが指定される `zoneadm clone` 操作の場合、または `-c profile.xml sysconfig` ファイルオプションが指定されない `attach` 操作の場合では、システム構成は必要ありません。

次のような制御が可能になります。

- `sysconfig configure` コマンドを使用して、対象の Oracle Solaris インスタンスを再構成 (構成解除してから構成) します。
 - `sysconfig configure` コマンドを使用して、対象の Oracle Solaris インスタンスを構成し、SCI ツールをコンソール上で開始します。


```
# sysconfig configure
```
 - `sysconfig configure` コマンドを使用して、大域ゾーンまたは非大域ゾーン内の構成解除済み Solaris インスタンスを構成します。


```
# sysconfig configure -c sc_profile.xml
```

このコマンドで既存の構成プロファイルを指定した場合、非対話型の構成が実行されます。このコマンドを使用して既存の構成プロファイルを指定しない場合、対話式システム構成 (SCI) ツールが実行されます。SCI ツールを使用すれば、対象の Oracle Solaris インスタンスに固有の構成情報を指定できます。
- 新しいシステム構成プロファイルを作成するには、`sysconfig create-profile` コマンドを使用できます。

`sysconfig` インタフェースについては、『[Oracle Solaris 11.3 システムのインストール](#)』の第 6 章、「[Oracle Solaris インスタンスの構成解除または再構成](#)」および [sysconfig\(1M\)](#) のマニュアルページに記載されています。

対話式システム構成ツール

対話式システム構成 (SCI) ツールを使用すると、新規にインストールされた Oracle Solaris インスタンスに構成パラメータを指定できます。

`-c profile.xml` オプションを指定しない `sysconfig configure` は、システムを構成解除したあと、SCI ツールを表示して管理者に確認してから `/etc/svc/profile/site/scit_profile.xml` にその構成を書き込みます。次に、このツールはこの情報を使用してシステムを構成します。

`sysconfig create-profile` は、管理者に確認してから、`/system/volatile/scit_profile.xml` に SMF プロファイルファイルを作成します。パラメータには、システムのホスト名、タイムゾーン、ユーザーと root のアカウント、ネームサービスが含まれます。

このツール内でナビゲートするには、次のようにします。

- 各画面の下部に一覧表示されたファンクションキーを使用して画面を移動し、ほかの操作を実行します。キーボードにファンクションキーがない場合や、キーが反応しない場合は、Esc キーを押します。画面下部の説明が変わり、ナビゲーションやほかの機能の各 Esc キーが表示されます。
- 選択を変更するか、または入力フィールド間で移動するには、上矢印/下矢印キーを使用します。

詳細は、『Oracle Solaris 11.3 システムのインストール』の第6章、「Oracle Solaris インスタンスの構成解除または再構成」および `sysconfig(1M)` のマニュアルページを参照してください。

ゾーン構成プロファイルの例

自動構成を持つ排
他的 IP ゾーン:

```
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/
service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/config-user">
    <instance enabled="true" name="default">
      <property_group type="application" name="root_account">
        <propval type="astring" name="login" value="root"/>
        <propval type="astring" name="password" value="$5$KeNRy1zU
$1qzy9rIsNloUhfVJFIWmVewE75aB5/EBA77kY7EP6F0"/>
        <propval type="astring" name="type" value="role"/>
      </property_group>
      <property_group type="application" name="user_account">
        <propval type="astring" name="login" value="admin1"/>
        <propval type="astring" name="password" value="$5$/g353K5q
$V8Koe/XuAeR/zpBvpLsgVIqPrvc.9z0hYFYoyoBkE37"/>
        <propval type="astring" name="type" value="normal"/>
        <propval type="astring" name="description" value="admin1"/>
        <propval type="count" name="gid" value="10"/>
        <propval type="astring" name="shell" value="/usr/bin/bash"/>
        <propval type="astring" name="roles" value="root"/>
        <propval type="astring" name="profiles" value="System
Administrator"/>
        <propval type="astring" name="sudoers" value="ALL=(ALL) ALL"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/timezone">
    <instance enabled="true" name="default">
      <property_group type="application" name="timezone">
        <propval type="astring" name="localtime" value="UTC"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/environment">
    <instance enabled="true" name="init">
      <property_group type="application" name="environment">
        <propval type="astring" name="LC_ALL" value="C"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/identity">
    <instance enabled="true" name="node">
      <property_group type="application" name="config">
        <propval type="astring" name="nodename" value="my-zone"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/keymap">
    <instance enabled="true" name="default">
      <property_group type="system" name="keymap">
        <propval type="astring" name="layout" value="US-English"/>
      </property_group>
    </instance>
  </service>
</service_bundle>
```

DNS と組み合わせない NIS を使用した静的構成を持つ排他的 IP ゾーン:

```

</service>
<service version="1" type="service" name="system/console-login">
  <instance enabled="true" name="default">
    <property_group type="application" name="ttymon">
      <propval type="astring" name="terminal_type" value="vt100"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="network/physical">
  <instance enabled="true" name="default">
    <property_group type="application" name="netcfg">
      <propval type="astring" name="active_ncp" value="Automatic"/>
    </property_group>
  </instance>
</service>
</service_bundle>

<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/
service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/config-user">
    <instance enabled="true" name="default">
      <property_group type="application" name="root_account">
        <propval type="astring" name="login" value="root"/>
        <propval type="astring" name="password" value="$5$m80R3zqK
$0x5XGubRJdi4zj0JzNSmVJ3Ni4opD0Gpxi2nK/GGzmC"/>
        <propval type="astring" name="type" value="normal"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/timezone">
    <instance enabled="true" name="default">
      <property_group type="application" name="timezone">
        <propval type="astring" name="localtime" value="UTC"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/environment">
    <instance enabled="true" name="init">
      <property_group type="application" name="environment">
        <propval type="astring" name="LC_ALL" value="C"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/identity">
    <instance enabled="true" name="node">
      <property_group type="application" name="config">
        <propval type="astring" name="nodename" value="my-zone"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/keymap">
    <instance enabled="true" name="default">
      <property_group type="system" name="keymap">
        <propval type="astring" name="layout" value="US-English"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/console-login">
    <instance enabled="true" name="default">
      <property_group type="application" name="ttymon">
        <propval type="astring" name="terminal_type" value="vt100"/>
      </property_group>
    </instance>
  </service>

```

```
</service>
<service version="1" type="service" name="network/physical">
  <instance enabled="true" name="default">
    <property_group type="application" name="netcfg">
      <propval type="astring" name="active_ncp" value="DefaultFixed"
/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="network/install">
  <instance enabled="true" name="default">
    <property_group type="application" name="
install_ipv4_interface">
      <propval type="astring" name="address_type" value="static"/>
      <propval type="net_address_v4" name="static_address" value="
10.10.10.13/24"/>
      <propval type="astring" name="name" value="net0/v4"/>
      <propval type="net_address_v4" name="default_route" value="
10.10.10.1"/>
    </property_group>
    <property_group type="application" name="
install_ipv6_interface">
      <propval type="astring" name="stateful" value="yes"/>
      <propval type="astring" name="stateless" value="yes"/>
      <propval type="astring" name="address_type" value="addrconf"/>
      <propval type="astring" name="name" value="net0/v6"/>
    </property_group>
  </instance>
</service>
<service version="1" type="service" name="system/name-service/
switch">
  <property_group type="application" name="config">
    <propval type="astring" name="default" value="files nis"/>
    <propval type="astring" name="printer" value="user files nis"/>
    <propval type="astring" name="netgroup" value="nis"/>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="system/name-service/
cache">
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/dns/client">
  <instance enabled="false" name="default"/>
</service>
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="example.net"/>
    <property type="host" name="ypservers">
      <host_list>
        <value_node value="192.168.224.11"/>
      </host_list>
    </property>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/client">
  <instance enabled="true" name="default"/>
</service>
</service_bundle>
```

NIS を使用した動的構成を持つ排他的 IP ゾーン

```
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/
service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
  <service version="1" type="service" name="system/config-user">
    <instance enabled="true" name="default">
      <property_group type="application" name="root_account">
        <propval type="astring" name="login" value="root"/>
        <propval type="astring" name="password" value="$5$Iq/.A.
K9$RQyt6RqsAY8TgnuxL9i0/84QwgIQ/nqcK8QsTQdvMy"/>
        <propval type="astring" name="type" value="normal"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/timezone">
    <instance enabled="true" name="default">
      <property_group type="application" name="timezone">
        <propval type="astring" name="localtime" value="UTC"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/environment">
    <instance enabled="true" name="init">
      <property_group type="application" name="environment">
        <propval type="astring" name="LC_ALL" value="C"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/identity">
    <instance enabled="true" name="node">
      <property_group type="application" name="config">
        <propval type="astring" name="nodename" value="my-zone"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/keymap">
    <instance enabled="true" name="default">
      <property_group type="system" name="keymap">
        <propval type="astring" name="layout" value="US-English"/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/console-login">
    <instance enabled="true" name="default">
      <property_group type="application" name="ttymon">
        <propval type="astring" name="terminal_type" value="sun-color"
/>
      </property_group>
    </instance>
  </service>
  <service version="1" type="service" name="system/name-service/
switch">
    <property_group type="application" name="config">
      <propval type="astring" name="default" value="files nis"/>
      <propval type="astring" name="printer" value="user files nis"/>
      <propval type="astring" name="netgroup" value="nis"/>
    </property_group>
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="system/name-service/cache">
    <instance enabled="true" name="default"/>
  </service>
  <service version="1" type="service" name="network/dns/client">
    <instance enabled="false" name="default"/>
  </service>
```

```
<service version="1" type="service" name="network/nis/domain">
  <property_group type="application" name="config">
    <propval type="hostname" name="domainname" value="special.
example.com"/>
    <property type="host" name="ypservers">
      <host_list>
        <value_node value="192.168.112.3"/>
      </host_list>
    </property>
  </property_group>
  <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/nis/client">
  <instance enabled="true" name="default"/>
</service>
</service_bundle>
```

非大域ゾーンへのログイン方法

このセクションでは、ゾーンへのログインに使用可能な方法について説明します。

ゾーンコンソールログイン

各ゾーンは、仮想コンソール `/dev/console` を保持します。コンソール上で操作を実行することを、コンソールモードと呼びます。ゾーンがインストール済み状態の場合、ゾーンへのコンソールログインを使用できます。ゾーンコンソールは、システム上のシリアルコンソールに非常に似ています。コンソールへの接続は、ゾーンをリブートしても持続します。コンソールモードと `telnet` などのログインセッションとの違いを理解するには、[75 ページの「リモートログイン」](#) を参照してください。

ゾーンコンソールへの接続には、`zlogin` コマンドと `-c` オプション、および `zonename` を使用します。ゾーンを稼働状態にする必要はありません。

また、`-d` オプションを使用することもできます。このオプションは、ゾーンが停止した場合に、ゾーンがコンソールから切断されるよう指定します。このオプションは、`-c` オプションとともにのみ指定できます。

ゾーン内部のプロセスが、コンソールを開いてメッセージを書き込むことができます。`zlogin -c` プロセスが終了すると、別のプロセスがコンソールにアクセスできるようになります。

役割によるアクセス制御 (RBAC) を使用中の場合、ゾーンコンソールへのアクセスには `solaris.zone.manage/zonename` の承認が必要になります。スラッシュ文字 (`/`) が前に付いた特定の `zonename` の接尾辞はオプションです。省略すると、承認はすべてのゾーンと一致します。

ブート時に対話式システム構成 (SCI) ツールを開始するには、次のように入力します。

```
root@test2:~# sysconfig configure -s
```

ユーザーログインの方法

ユーザー名を使ってゾーンにログインする場合は、`zlogin` コマンドと `-l` オプション、ユーザー名、および `zonename` を使用します。たとえば、大域ゾーンの管理者は、`zlogin` に `-l` オプションを指定することで、通常のコマンドラインユーザーとして非大域ゾーンにログインできます。

```
global# zlogin -l user zonename
```

ユーザー `root` でログインするには、オプションを指定せずに `zlogin` を使用します。

フェイルセーフモード

ログインで問題が発生し、`zlogin` コマンドまたは `zlogin` コマンドと `-c` オプションを使用してゾーンにアクセスできない場合、代替手段が存在します。`zlogin` コマンドと `-S (safe)` オプションを使用することで、ゾーンに入ることができます。このモードは、ほかのログイン方法が成功しなかったときに、損傷を受けたゾーンを復元する場合にのみ使用してください。この最小環境では、ゾーンログインが失敗した理由を診断できる場合があります。

リモートログイン

Oracle Solaris では、デフォルトでのセキュリティー保護 (SBD) 機能が自動的に実装されます。この機能では、`ssh` が Oracle Solaris システムへの有効な唯一のリモートログインです。`ssh` を使用して非大域ゾーンに入ります。`rlogin` や `telnet` などのほかのリモートログインサービスは、セキュリティー保護されておらず、ネットワークが不正アクセスにさらされる可能性があります。詳細は、[ssh\(1\)](#) のマニュアルページを参照してください。

対話型モードと非対話型モード

`zlogin` コマンドを使ってゾーンにアクセスし、ゾーン内でコマンドを実行する方法がさらに 2 つ存在します。その方法が、対話型モードおよび非対話型モードです。

対話型モード

対話型モードでは、ゾーン内部で使用する新しい仮想端末が割り当てられます。コンソールデバイスへの排他的なアクセスが許可されるコンソールモードとは異なり、対話型モードでは、いつでも任意の数の `zlogin` セッションを開くことができます。対話型モードが有効になるのは、発行するコマンドが含まれていない場合です。エディタなどの端末デバイスを必要とするプログラムは、このモードで正常に動作します。

RBAC を使用中の場合、対話型ログインには、ゾーンに対する `solaris.zone.login/zonename` の承認が必要です。ゾーン内では、パスワード認証が実行されます。

非対話型モード

非対話型モードは、ゾーンを管理するシェルスクリプトを実行する場合に使用します。非対話型モードでは、新しい仮想端末は割り当てられません。ゾーン内部で実行されるコマンドを指定すると、非対話型モードが有効になります。

非対話型ログインの場合、またはパスワード認証を省略する場合は、`solaris.zone.manage/zonename` の承認が必要です。

◆◆◆ 第 5 章

非大域ゾーンへのログイン

この章では、インストール済みのゾーンの構成を完了して大域ゾーンからゾーンにログインし、ゾーンを停止処理する手順について説明します。また、この章では、`zonename` コマンドを使用して現在のゾーンの名前を出力する方法についても説明します。

ゾーンへのログイン処理の概要については、[第4章「非大域ゾーンへのログインについて」](#)を参照してください。

初期のゾーンブートおよびゾーンログイン手順

タスク	説明	手順
内部構成を実行するか、またはゾーンを構成解除します。	システム構成は、テキストのユーザーインターフェースを使用して対話的に行うことができ、またプロファイルを使用して非対話的に行うこともできます。また、Solaris インスタンスを構成解除するために、 <code>sysconfig</code> ユーティリティも使用します。	『Oracle Solaris 11.3 システムのインストール』の第6章、「Oracle Solaris インスタンスの構成解除または再構成」および <code>sysconfig(1M)</code> のマニュアルページを参照してください。
ゾーンにログインします。	ゾーンへのログインには、コンソールを使用する、対話型モードを使って仮想端末を割り当てる、またはゾーン内で実行するコマンドを指定する方法があります。実行するコマンドを指定する場合、仮想端末は割り当てられません。ゾーンへの接続が拒否された場合は、フェイルセーフモードを使用してログインすることもできます。	78 ページの「ゾーンへのログイン」
非大域ゾーンから抜けます。	非大域ゾーンへの接続を切り離します。	81 ページの「非大域ゾーンから抜ける方法」
ゾーンを停止処理します。	<code>shutdown</code> ユーティリティまたはスクリプトを使用して、ゾーンを停止処理します。	82 ページの「 <code>zlogin</code> を使用してゾーンを停止処理する方法」

タスク	説明	手順
ゾーン名を出力します。	現在のゾーンの名前を出力します。	83 ページの「現在のゾーンの名前を出力する」

ゾーンへのログイン

zlogin コマンドを使用して、大域ゾーンから稼働状態または準備完了状態にある任意のゾーンへログインします。詳細は、zlogin(1) のマニュアルページを参照してください。

次の手順で説明されているように、ゾーンへのログインはさまざまな方法で実行できます。75 ページの「リモートログイン」で説明されているように、リモートでログインすることも可能です。

▼ 構成プロファイルを作成する方法



注意 - 必要なすべてのデータを提供する必要がありますことに注意してください。データが欠落したプロファイルを提供すると、ゾーンはデータが欠落した状態で構成されます。この構成によって、ユーザーがログインできなくなったり、ネットワークを実行できなくなったりする可能性があります。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. sysconfig ツールを使用してプロファイルを作成します。

■ 排他的 IP ゾーンの場合:

```
# sysconfig create-profile -o /path/sysconf.xml
```

■ 共有 IP ゾーンの場合:

```
# sysconfig create-profile -o /path/sysconf.xml -g location,identity,naming_services,users
```

3. ゾーンのインストール、クローン、または接続の操作中に、この作成したプロファイルを使用します。

```
# zoneadm -z my-zone install -c /path/sysconf.xml
```

構成ファイルが使用されている場合、システムは最初の `zlogin` 時にコンソール上で対話式システム構成 (SCI) ツールを開始しません。ファイルの引数は絶対パスで指定する必要があります。

▼ ゾーンコンソールにログインして初期ゾーン構成を行う方法

`config.xml` ファイルが `zoneadm clone`、`attach`、または `install` コマンドに渡されると、この構成ファイルがシステムを構成するために使用されます。`config.xml` ファイルが `clone`、`attach`、または `install` の操作中に提供されなかった場合、ゾーンの最初のブートではコンソール上で SCI ツールを開始します。

構成情報用の最初のプロンプトが表示されないことを避けるために、ゾーンが 2 回目のセッションでブートされる前に、`zlogin` が実行されるように 2 つの端末ウィンドウを使用することをお勧めします。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. `zlogin` コマンドを `-c` オプションと `my-zone` などのゾーン名とともに使用します。

```
global# zlogin -c my-zone
```

3. 別の端末ウィンドウからゾーンをブートします。

```
global# zoneadm -z my-zone boot
```

次のような内容が、`zlogin` 端末ウィンドウに表示されます。

```
[NOTICE: Zone booting up]
```

4. 新たにインストールしたゾーン用の構成パラメータに関する一連の質問に回答します。パラメータには、システムのホスト名、タイムゾーン、ユーザーと `root` アカウント、およびネームサービスが含まれます。SCI ツールは、デフォルトでは `/system/volatile/scit_profile.xml` に SMF プロファイルファイルを生成します。

注意事項 最初の SCI 画面が表示されない場合は、`Ctrl+L` を押してこの SCI 画面をリフレッシュできます。

▼ ゾーンコンソールへのログイン方法

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. **管理者になります。**

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

2. **zlogin コマンドを -c オプション、-d オプション、および my-zone などのゾーンの名前とともに使用します。**

```
global# zlogin -c -d my-zone
```

構成が実行されていない場合、zlogin コマンドを -c オプションとともに使用すると SCI ツールが開始されます。

3. **ゾーンコンソールが表示されたら、root でログインし、Return キーを押します。プロンプトが表示されたら root のパスワードを入力します。**

```
my-zone console login: root
Password:
```

▼ 対話型モードを使用してゾーンにアクセスする方法

対話型モードでは、ゾーン内部で使用する新しい仮想端末が割り当てられます。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. **管理者になります。**

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

2. **大域ゾーンからゾーン (例: my-zone) にログインします。**

```
global# zlogin my-zone
```

次のような情報が表示されます。

```
[Connected to zone 'my-zone' pts/2]
Last login: Wed Jul  3 16:25:00 on console
```

3. **exit と入力して、接続を閉じます。**

次のようなメッセージが表示されます。

```
[Connection to zone 'my-zone' pts/2 closed]
```

▼ 非対話型モードを使用してゾーンにアクセスする方法

ゾーン内部で実行されるコマンドを指定すると、非対話型モードが有効になります。非対話型モードでは、新しい仮想端末は割り当てられません。

コマンドおよびコマンドの処理対象のファイルは、いずれも NFS 上に存在してはならないことに注意してください。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. 大域ゾーンから my-zone ゾーンにログインして、コマンド名を入力します。

ここではコマンド `zonename` を使用します。

```
global# zlogin my-zone zonename
```

次の出力が表示されます。

```
my-zone
```

▼ 非大域ゾーンから抜ける方法

● 非大域ゾーンへの接続を切り離すには、次のいずれかの方法を使用します。

■ ゾーンの非仮想コンソールを終了するには、次の操作を行います。

```
zonename# exit
```

■ ゾーンの仮想コンソールへの接続を切り離すには、次のようにチルダ (~) 文字とピリオドを使用します。

```
zonename# ~.
```

画面には、次のようなメッセージが表示されます。

```
[Connection to zone 'my-zone' pts/6 closed]
```

注記 - また、ssh のデフォルトのエスケープシーケンスも ~ で、これは ssh セッションを終了させます。ssh を使用してリモートからサーバーにログインする場合は、~~. を使用してゾーンを抜けます。

参照 `zlogin` コマンドのオプションの詳細は、[zlogin\(1\)](#) のマニュアルページを参照してください。

▼ フェイルセーフモードを使用してゾーンに入る方法

ゾーンへの接続が拒否された場合、`zlogin` コマンドと `-s` オプションを使用して、ゾーン内の最小環境に入ることができます。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. 大域ゾーンから、`zlogin` コマンドと `-s` オプションを使用してゾーン (例: `my-zone`) にアクセスします。

```
global# zlogin -s my-zone
```

▼ `zlogin` を使用してゾーンを停止処理する方法

注記 - 大域ゾーンで `init 0` を実行して Oracle Solaris システムの完全な停止処理を実行すると、システム上のそれぞれの非大域ゾーンでも `init 0` が実行されます。`init 0` は、ローカルユーザーとリモートユーザーに対してシステムが停止する前にログオフするよう警告しません。

ゾーンを正しく停止処理するには、次の手順を実行します。停止処理スクリプトを実行せずにゾーンを停止する方法については、[58 ページの「ゾーンの停止方法」](#)を参照してください。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

1. 管理者になります。

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

2. 停止処理を行うゾーン (例: `my-zone`) にログインし、ユーティリティーの名前として `shutdown` を、状態として `init 0` を指定します。

```
global# zlogin my-zone shutdown -i 0
```

サイトによっては、特定の環境に合わせた独自の停止処理スクリプトが存在する場合があります。

サービスの有効化

ゾーン内にある個々のサービスを有効または無効にできます。

現在のゾーンの名前を出力する

zonename(1)のマニュアルページに説明されているように、zonename コマンドにより現在のゾーンの名前が出力されます。次に、大域ゾーン内で zonename を使用した場合の出力例を示します。

```
# zonename
global
```


ライブゾーン再構成

非大域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。

ライブゾーン再構成について

次のタスクを行う場合に、ライブゾーン再構成を使用します。

- 現在のライブゾーン構成のレポートを作成し検査する
- ライブゾーン構成を変更する
- ライブゾーン構成に変更を適用する

一時的な変更を行うことも、永続的な変更を行うこともできます。標準ツール `zonecfg` および `zoneadm` は、ライブゾーン再構成の管理に使用されます。一時的な変更は、次のリブートまで有効です。変更を永続的構成に適用する場合は、リブートする必要はありません。

次の構成変更を行なった場合、ゾーン内でのサービスの可用性が中断することはありません。

- リソース制御の変更
- ネットワーク構成の変更
- CPU リソースプールの変更
- ファイルシステムの追加または削除
- 仮想および物理デバイスの追加または削除

一時的な変更について

ライブモードでのパラメータの変更は、確定した直後から有効になります。zonecfg コマンドで行われた一時的な変更は、次のゾーンのリブートまで有効です。ライブモードは実行中のゾーンにのみ使用可能です。

構成に対する変更について

zonecfg コマンド編集モードを使用して、ゾーンに対して構成の変更を行います。格納済みの永続的構成でも、実行中のライブ構成でも変更できます。zonecfg コマンドは、ライブ再構成で使用する場合、次の編集モードをサポートします。

デフォルトモード 安定したストレージに格納された永続的ゾーン構成を作成、変更、および一覧表示します。デフォルトの編集モードで変更したパラメータは、変更が行われた時点では実行中のゾーンに影響しません。デフォルトの編集モードは、ゾーン構成を維持するための主要な方法です。このモードには下位互換性があります。ゾーンは、変更を有効にするには、次の方法のいずれかで再構成する必要があります。

- zoneadm apply
- zoneadm reboot

ライブモード 実行中のライブ構成を取得、検査、および編集します。ライブモードは実行中のゾーンにのみ使用可能です。ライブモードで変更されたパラメータは、確定した直後に有効になります。適用された変更は、ゾーンが次回リブートするまで有効です。ライブモードは -r オプションで有効になります。このオプションは永続的構成ではなくライブ構成を取得および編集するために使用されます。

```
# zonecfg -z zonename -r
```

構成への変更の適用について

zoneadm apply コマンドを使用すると、自身で行なった変更をライブまたは永続的ゾーン構成に適用できます。永続的な変更にするためにリブートする必要はありません。永続的な変更にする前に、-n オプションを使用することによって試行を実行できます。apply サブコマンドに次のオプションを使用できます。

- n 「実行なし」オプション(-n)を使用した試行。この試行(予行演習モードとも呼ばれます)は、実際の再構成を使用しますが、実行中のゾーンに変更は適用されません。予行演習モードを使用して、実際の再構成によって実行されるアクションを確認します。
- q 静寂モード。このモードは、すべてのシステムメッセージを抑制し、ステータスコードだけを返します。

ライブゾーン再構成の例

このセクションの例を使用して、一般的なゾーン再構成タスクを実行します。

▼ 実行中のゾーンのライブ構成を検査する方法

すべてのブランドは構成を検査できます。

1. 管理者になります。

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

2. ゾーン *my-zone* のライブゾーン構成に関する情報を表示します。

```
# zonecfg -z my-zone -r info
```

3. (オプション) ライブ構成をエクスポートします。

```
# zonecfg -z my-zone -r export -f exported.cfg
```

すべてのブランドは構成をエクスポートできます。

▼ 予行演習を使用して可能な構成を表示する方法

予行演習は、試行とも呼ばれます。

1. 管理者になります。

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

2. `zonecfg` コマンドを使用して変更を構成します。

```
# zonecfg -z my-zone -r
```

3. `-n` オプションを指定して `commit` サブコマンドを使用し、実際の再構成によって実行されるアクションを表示します。

```
zonecfg:my-zone> commit -n
```

▼ 永続的な構成変更を行い、変更を適用する方法

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. `zone1` というゾーンに変更を加え、変更を適用します。

```
# zonecfg -z zone1 "set cpu-shares=4;clear pool;add anet;..."
# zoneadm -z zone1 apply
zone1: Checking: set property cpu-shares=4
zone1: Checking: clear property pool
zone1: Checking: add anet linkname=myanet0
zone1: Applying changes
```

▼ 実行中のゾーンに anet リソースを一時的に追加する方法

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. 追加を行い、変更を適用します。

```
# zonecfg -z zone1 -r "add anet;set linkname=anet1;set lower-link=net1;end;commit"
zone1: Checking: add anet linkname=anet1
zone1: Applying changes
```

例における `commit` サブコマンドは不要です。 `zonecfg` コマンドは、コマンドの終了時に変更を確定します。

▼ 実行中のゾーンに一時的な変更を加える方法

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. ディスクの追加など、構成の変更を行います。

```
# zonecfg -z zone1 -r 'add device; set match=/dev/rdisk/c1t3d0*;end; \
  add device; set match=/dev/dsk/c1t3d0*; end;'
zone1: Checking: Adding device match=/dev/rdisk/c1t3d0*
zone1: Checking: Adding device match=/dev/dsk/c1t3d0*
zone1: Applying the changes
```

- 追加したディスクが不要になった場合は、ゾーンを永続的構成に戻します。

```
# zoneadm -z zone1 apply
zone 'zonename': Checking: Removing device match=/dev/rdisk/c1t3d0*
zone 'zonename': Checking: Removing device match=/dev/dsk/c1t3d0*
zone 'zonename': Applying changes
```

▼ 一時的な変更の確定中に障害から回復する方法

ライブ構成の編集に、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。この競合が発生すると、commit サブコマンドはエラーを返します。構成をリロードして更新されたバージョンを表示し、編集を行うことができます。

- 管理者になります。

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

- 次の手順で reload を使用して、障害から回復します。

```
# zonecfg -z zone1 -r
zonecfg:zone1> add anet;set linkname=anet1;set lower-link=net1;end
zonecfg:zone1> commit
zone1: error: Failed to commit. The live configuration of the zone
'zone1' changed externally.
zonecfg:zone1> reload
zonecfg:zone1> add anet;set linkname=anet1;set lower-link=net1;end
zonecfg:zone1> commit
zone1: Checking: add anet linkname=anet1
zone1: Applying changes
```


Oracle Solaris ゾーンの移行および変換

この章では、既存の solaris ブランドゾーンとそのデータをソース Oracle Solaris 11 システムからターゲット Oracle Solaris 11 システムに移行する方法について説明します。

この章ではまた、ゾーンの変換についても説明します。ゾーンの変換は、次のプロセスで使用されます。

- 大域ゾーンから非大域ゾーンへの変換 (物理から仮想 (P2V) への変換とも呼ばれます)
- 非大域ゾーンから大域ゾーンへの変換

移行では、アーカイブを使用したデータ転送が必要になることがあります。これらのアーカイブは、Oracle Solaris 統合アーカイブでも、tar、cpio、pax などの従来のアーカイブタイプでもかまいません。Oracle Solaris 11 の場合は、アーカイブを使用したゾーンの変換をゾーン移行戦略の一部にできます。

ゾーンの移行は、サーバーの統合、ワークロードの負荷調整、および障害回復のために使用されます。

solaris ブランドゾーンの移行について

ゾーンの移行は、既存のゾーン構成とデータを別のホストシステムに転送するプロセスです。ソースゾーンは停止されて移動され、ターゲットホスト上でリポートされます。大域ゾーンと非大域ゾーンの両方を移行できます。ゾーンの移行およびゾーンの移行タイプの詳細は、『[Oracle Solaris ゾーンを紹介](#)』の「[ゾーンの移行について](#)」を参照してください。

注記 - solaris10、solaris、および solaris-kz ゾーンはすべて、移行をサポートしています。移行の要件については、そのゾーンブランドのゾーンのドキュメントを参照してください。

Oracle Solaris 11 の場合、移行は共有ストレージ (ZOSS) 上のゾーンを使用してデータを転送します。ZOSS を使用した移行には、ゾーンのレポートとほぼ同じ時間がかか

ります。ZOSSの詳細は、[第13章「共有ストレージでの Oracle Solaris ゾーンの使用開始」](#)を参照してください。

ZOSSが使用できない場合は、ゾーンの変換によって作成されたアーカイブを移行データ転送戦略の一部として使用できます。[92 ページの「ゾーンの変換および Oracle Solaris ゾーンの変換について」](#)を参照してください。

ゾーンの変換および Oracle Solaris ゾーンの変換について

ゾーンの変換は、Oracle Solaris 大域ゾーンまたは非大域ゾーンのアーカイブを作成し、それを Oracle Solaris 大域ゾーンまたは非大域ゾーンとして配備するプロセスです。ソースゾーンのタイプ(大域または非大域)は、配備されるゾーンタイプ(大域または非大域)と同じ場合と異なる場合があります。

ゾーンは、次のように変換できます。

- 大域ゾーンを非大域ゾーンに変換できます。このプロセスは、物理から仮想への変換(P2V変換)とも呼ばれます。
- 非大域ゾーンを非大域ゾーンに変換できます。このプロセスは、仮想から仮想への変換(V2V変換)と呼ばれる場合があります。
- 非大域ゾーンを大域ゾーンに変換できます。このプロセスは、仮想から物理への変換と呼ばれる場合があります。

Oracle Solaris 10 システム上では、アーカイブを使用すると、アップグレードされたホストで Oracle Solaris 10 native ゾーンを solaris10 ブランドゾーンに変換できます。

注記 - Oracle Solaris は、複数のタイプの仮想化をサポートしています。たとえば、論理ドメインとゾーンは、どちらも仮想化のタイプです。論理ドメインのゾーンへの変換は、V2V変換のタイプです。ゾーンの変換の観点からは、それは大域から非大域への変換(P2V変換)です。Oracle Solaris 11 以降のリリースでのゾーンの変換のモデルは、物理対仮想ではなく、大域対非大域の観点から説明されます。

ゾーンの変換では、次のアーカイブタイプがサポートされています。

- 統合アーカイブ。[93 ページの「ゾーンの変換のための Oracle Solaris 統合アーカイブの使用」](#)を参照してください。
- 従来のアーカイブ。[93 ページの「ゾーンの変換のための従来のアーカイブの使用」](#)を参照してください。

統合アーカイブは、Oracle Solaris 11.2 以降を実行しているホストシステムのゾーンの変換に推奨されます。11.2 より前の Oracle Solaris 11 リリースを実行しているホストシステムの場合は、従来のアーカイブを使用してください。

注記 - 仮想から物理への変換は、統合アーカイブを使用している場合にのみサポートされます。統合アーカイブの詳細は、『[Oracle Solaris 11.3 でのシステム復旧とクローン](#)』を参照してください。

大域ゾーンを非大域ゾーンに変換する前に、大域ゾーンで実行されているソフトウェアが非大域ゾーンと互換性があることを確認できます。[zonep2vchk\(1M\)](#) ツールは、非大域ゾーンへの変換前に大域ゾーンの構成を評価します。

ゾーンの変換のための Oracle Solaris 統合アーカイブの使用

`zonecfg` および `zoneadm` を使用すると、統合アーカイブファイルから直接、新しいゾーンを構成およびインストールできます。統合アーカイブファイルには、ゾーン構成とゾーンデータの両方が含まれています。ターゲットとなる移行先システムで、ゾーンをアーカイブから構成およびインストールできます。

このリリースの場合、システムおよびゾーンのクローニングや回復のための使用方法を含む統合アーカイブの詳細は、『[Oracle Solaris 11.3 でのシステム復旧とクローン](#)』を参照してください。

ゾーンの変換のための従来のアーカイブの使用

従来のアーカイブには、`zfs send`、`cpio`、または `pax` で作成されたアーカイブファイルが含まれています。ゾーンの変換のための従来のアーカイブは、次の場合にのみ考慮するようにしてください。

- ソースシステムが 11.2 より前の Oracle Solaris リリースを実行している場合。
- 共有ストレージ (ZOSS) 上のゾーンが使用できない場合。

詳細は、`zfs`、`cpio`、または `pax` のマニュアルページを参照してください。

zonep2vchk ツールを使用した大域から非大域への変換準備

[zonep2vchk\(1M\)](#) ツールは、非大域ゾーンへの変換前に大域ゾーンの構成を評価します。このツールの主なドキュメントは、[zonep2vchk\(1M\)](#) のマニュアルページです。

zonep2vchk ツールについて

zonep2vchk は、大域ゾーン (または物理インスタンス) を、それが非大域ゾーン (または仮想インスタンス) に変換されるときに評価します。物理から仮想への (P2V) 変換プロセスは、大域ゾーン (ソース) をアーカイブすること、およびそのあとそのアーカイブを使用して非大域ゾーン (ターゲット) をインストールすることで構成されます。zonep2vchk ユーティリティーは、`0` の実効ユーザー `id` で実行する必要があります。

このユーティリティーは次のことを行います。

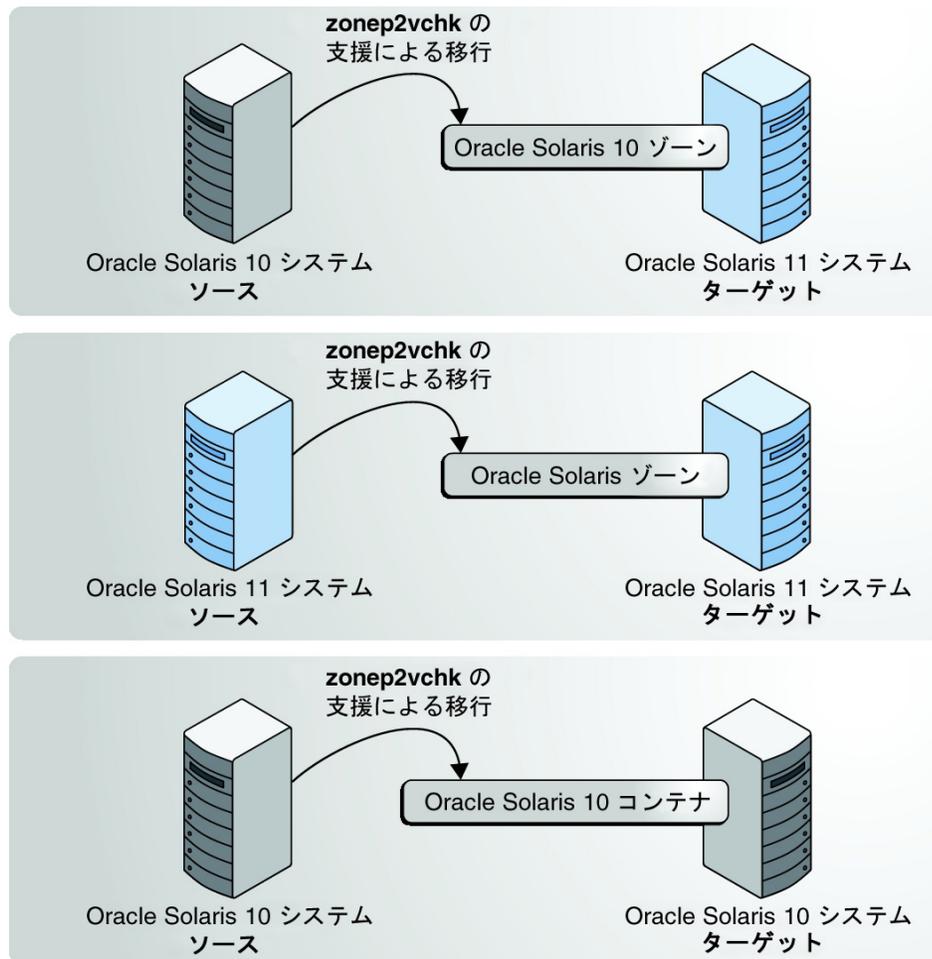
- ソースシステムの構成内の問題領域を特定します
- 必要になる手動の再構成作業を最小限に抑えます
- Oracle Solaris 10 システムイメージと Oracle Solaris 11 システムイメージの両方の Oracle Solaris 11 リリース上のゾーンへの変換をサポートします。
- 元のシステムイメージ内の複雑なネットワーク構成 (複数の IP インタフェース、IP マルチパス、VLAN など) をサポートします

このツールを使用すると、Oracle Solaris 11 または Oracle Solaris 10 大域ゾーンをこのリリース上の非大域ゾーンに変換するのに役立ちます。このツールにより、次の変換のタイプが得られます。

- Oracle Solaris 11 システムを `solaris` ブランドゾーンに変換する
- Oracle Solaris 10 システムを `solaris10` ブランドゾーンに変換する

Oracle Solaris 11 ターゲットシステムの場合は、ソースシステム上のネットワークリソースごとの `zonecfg` 出力に `anet` リソース (VNIC) が含まれます。デフォルトでは、Oracle Solaris 11 システム上の非大域ゾーンに Oracle Solaris 11 システムまたは Oracle Solaris 10 システムのいずれかを移行する場合のネットワークタイプは排他的 IP です。

図 2 zonep2vchk ユーティリティ



解析のタイプ

基本解析 (-b オプション) は、大域から非大域への変換によって影響を受ける可能性のある使用中の Oracle Solaris 機能をチェックします。

静的解析 (-s オプション) は、ゾーン内で機能しない可能性のあるシステムおよびライブラリコールがないかどうかバイナリを検査します。

実行時解析 (-r オプション) は、現在実行中のアプリケーションに、ゾーン内で機能しない可能性がある操作がないかどうかを検査します。

生成される情報

次の 2 つの主なカテゴリの情報が解析によって提示されます。

- 特定のゾーンの構成または大域ゾーン内の構成変更によって対処できる問題
- ゾーン内で実行できない機能の特定

たとえば、アプリケーションがシステムクロックを設定する場合、適切な特権をゾーンに追加することによって有効にできますが、アプリケーションがカーネルメモリーにアクセスする場合、それはゾーン内部では許可されることはありません。出力では、これらの 2 つのクラスの問題を区別します。

このユーティリティーは、デフォルトでは人間が読める形式のメッセージを出力します。メッセージを機械解析可能な形式で出力するには、-P オプションが使用されます。使用可能なオプションやコマンドの呼び出しと出力の詳細は、[zonep2vchk\(1M\)](#) のマニュアルページを参照してください。

solaris ブランドゾーンの移行およびゾーンの変換の要件

solaris ブランドゾーンに対するゾーンの移行とゾーンの変換の両方には、次の要件があります。

- ターゲットシステム上の大域ゾーンは、元のソースホスト上のリリース以上の Oracle Solaris 11 リリースを実行している必要があります。
- ゾーンが正常に動作することを保証するには、元のソースホストにインストールされている必須オペレーティングシステムパッケージと同じかまたはそれ以降のバージョンのものが、ターゲットシステムにインストールされている必要があります。
サードパーティー製品のパッケージなど、ほかのパッケージは異なっていてもかまいません。

注記 - solaris10 ブランドゾーンと solaris-kz ブランドゾーンには、ゾーンの変換およびゾーンの移行に対するブランド固有の追加の要件があります。solaris10 ブランドゾーンについては、『[Oracle Solaris 10 ゾーンの実行と使用](#)』の第 2 章、「[Oracle Solaris 10 システムの評価とアーカイブの実行](#)」を参照してください。solaris-kz ブランドゾーンについては、『[Oracle Solaris カーネルゾーンの実行と使用](#)』を参照してください。

solaris ゾーンを移行するためのタスク

このセクションでは、solaris ブランドゾーンの移行の管理および関連するタスクについて説明します。次の内容について説明します。

- [97 ページの「孤立したゾーンブート環境について」](#)
- [99 ページの「非大域ゾーンの別のマシンへの移行について」](#)
- [104 ページの「使用できないマシンからのゾーンの移行について」](#)

孤立したゾーンブート環境について

ネイティブな非大域ゾーンは、別のホストに移行されると、既存の大域ゾーンに関連付けられていないゾーンブート環境を累積できます。非大域ゾーンブート環境は、孤立したブート環境と呼ばれます。

どの大域ゾーンにも関連付けられていないゾーンブート環境が選択された場合、そのゾーンブート環境はクローニングされます。このクローニングされたブート環境が更新され、その親はアクティブな大域ブート環境に再設定されます。孤立したブート環境は引き続き存在します。

次の `-x` オプションは、ゾーンの `zoneadm attach` 中にゾーンブート環境を管理するために使用されます。

- `destroy-orphan-zbes`
- `force-zbe-clone`
- `deny-zbe-clone`

注記 - `force-zbe-clone` と `deny-zbe-clone` は、相互に排他的なオプションです。オプション `attach-last-booted-zbe` および `destroy-orphan-zbes` は、`force-zbe-clone` または `deny-zbe-clone` オプションとともに使用できます。

`destroy-orphan-zbes`

このオプションは、どの大域ゾーンにも関連付けられていないすべてのゾーンブート環境を破棄します。

`attach` 中に、孤立したすべてのゾーンブート環境を破棄するには、次を使用します。

`-x destroy-orphan-zbes`

force-zbe-clone

選択されたゾーンブート環境が現在アクティブな大域ゾーンブート環境に関連付けられている場合は、選択されたゾーンブート環境が接続されます。この動作は、`-x force-zbe-clone` を使用して変更できます。

別の大域ゾーンブート環境を選択するには、次を使用します。

```
-x force-zbe-clone
```

deny-zbe-clone

デフォルトでは、次のいずれかが当てはまる場合は、選択されたゾーンブート環境がクローニングされ、そのクローンが接続されます。

- 選択されたゾーンブート環境が別の大域ゾーンブート環境に関連付けられている。
- 選択されたゾーンブート環境が、いずれの大域ゾーンブート環境にも関連付けられていない孤立したブート環境である。

リブート時にアクティブとマークされた孤立したゾーンブート環境を接続しようとする、`deny-zbe-clone` を使用する接続操作が失敗します。このオプションを使用するには、ゾーンブート環境は非アクティブとマークされている必要があります。ブート環境を非アクティブにマークするには、*Oracle Solaris 12* ブート環境の作成と管理の第4章「ブート環境の管理」を参照してください。

元のゾーンブート環境は引き続き存在します。

クローンを拒否して接続するには、次を使用します。

```
-x deny-zbe-clone
```

beadm list の変更

孤立したブート環境は、`0` フラグを使用して `beadm` によって `active` 列内に示されません。次の出力では、`solaris-0`、`solaris-1`、および `solaris-2` が孤立しています。

```
root@t1: # beadm list
BE          Active Mountpoint Space   Policy Created
-----
solaris-0 !RO    -           3.40M  static 2014-01-17 15:04
solaris-1 !RO    -           3.85M  static 2014-02-09 18:11
solaris-2 !RO    -           39.0K  static 2014-02-10 04:54
solaris-3 !R     -           2.19G  static 2014-02-12 16:56
solaris-4 !R     -           3.0K   static 2014-02-19 19:18
solaris-5 NR    /           2.43G  static 2014-03-20 04:38
```

孤立しているとして示されているブート環境は、`beadm destroy` で破棄できます。孤立したすべてのブート環境を破棄するには、`-0` を `beadm destroy` とともに使用します。

ブート環境の詳細は、『Oracle Solaris 11.3 ブート環境の作成と管理』および [beadm\(1M\)](#) のマニュアルページを参照してください。

非大域ゾーンの別のマシンへの移行について

▼ 共有ストレージを使用して非大域ゾーンを移行する方法

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

ゾーンのストレージが `rootzpool` リソースを使用して構成されており、データセットリソースがなく、かつオプションで1つ以上の `rpool` リソースを含んでいる場合は、移行が迅速かつ単純になります。この手順では、ソースシステムとターゲットホスト（この手順の例では `host1` と `host2`）の両方から、`rootzpool` および `zpool` リソースで参照されるストレージにアクセスできる必要があります。

1. 管理者になります。

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

2. 移行先ホストでゾーンを構成します。

```
source# zonecfg -z zonename export -f /net/hostname/zonename.cfg
target# zonecfg -z zonename -f /net/hostname/zonename.cfg
```

例:

```
host1# zonecfg -z my-zone export -f /net/my-host/my-zone.cfg
host2# zonecfg -z my-zone -f /net/my-host/my-zone.cfg
```

3. ゾーンを停止処理します。

```
source# zoneadm -z zonename shutdown
```

例:

```
host1# zoneadm -z my-zone shutdown
```

4. ソースホストからゾーンを切り離します。

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

例:

```
host1# zoneadm -z my-zone detach
```

5. ゾーンを移行先ホストに接続します。
オプション (-u、-U) が必要になる場合があります。

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

例:

```
host2# zoneadm -z my-zone attach
```

6. ゾーンをブートします。

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

例:

```
host2# zoneadm -z my-zone boot
```

▼ 統合アーカイブを使用して非大域ゾーンを変換する方法

1. 管理者になります。

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

2. 復旧用アーカイブを作成します。

ソースシステムでこの手順を使用して、移行するゾーンの回復用アーカイブを作成します。

```
source# archiveadm create -r -z zonename archive-name
```

例:

```
host1# archiveadm create -r -z my-zone /net/server/my-zone-archive.uar
```

3. ソースの大域ゾーンにあるゾーンをアンインストールするか、またはそのゾーンの **autoboot** を **false** に設定します。

```
source# zonecfg -z zonename set autoboot=false
```

4. アーカイブからターゲットとなる移行先の大域ゾーンを構成します。

```
target# zonecfg -z zonename create -a /net/server/zonename.uar
```

例:

```
host2# zonecfg -z my-zone create -a /net/server/my-zone-archive.uar
```

5. アーカイブを使用してゾーンをインストールします。

```
target# zoneadm -z zonename install -a archive-name
```

6. 移行したゾーンをブートします。

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

参照 統合アーカイブの作成および配備の詳細は、『[Oracle Solaris 11.3 でのシステム復旧とクローン](#)』の第2章、「[統合アーカイブの操作](#)」を参照してください。

▼ ZFS アーカイブを使用して別のホスト上の非大域ゾーンを変換したり、配備したりする方法

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を持つユーザーである必要があります。

注記 - Oracle Solaris の将来のリリースでは、ZFS アーカイブやその他の従来のアーカイブタイプに対するサポートが削除される可能性があります。Oracle Solaris 11.2 以降を実行しているシステムのアーカイブゾーン変換戦略として統合アーカイブをお勧めします。

この例では、zfs コマンドを使用して非大域ゾーンのアーカイブを作成する方法について説明します。このアーカイブはそのあと、別のシステムに接続されて配備されます。

この例では、ソースホストとターゲットホスト上の管理者が一時ファイルストレージ用の共有 NFS サーバーにアクセスできることを前提にしています。共有の一時領域が利用できない場合、scp セキュアコピー (リモートファイルのコピープログラム) などのほかの手段を使用すると、ソースマシンとターゲットマシン間でファイルをコピーできます。パスワードまたはパスフレーズが認証に必要な場合、scp プログラムはそれらの入力を求めます。

1. 管理者になります。

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

2. 移行するゾーン (この手順では my-zone) を停止処理します。

```
source# zoneadm -z my-zone shutdown
```

3. (オプション) ゾーンを切り離します。

```
source# zoneadm -z my-zone detach
```

切り離されたゾーンは、現在、構成済みの状態にあります。次回の大域ゾーンのブート時に、ゾーンは自動的にブートされません。

4. ゾーンの構成をエクスポートします。

```
source# mkdir /net/server/zonearchives/my-zone
source# zonecfg -z my-zone export> /net/server/zonearchives/my-zone/my-zone.zonecfg
```

5. gzip ZFS アーカイブを作成します。

```
source# zfs list -H -o name /zones/my-zone
rpool/zones/my-zone
source# zfs snapshot -r rpool/zones/my-zone@v2v
source# zfs send -rc rpool/zones/my-zone@v2v | gzip > /net/server/zonearchives/my-zone/my-zone.zfs.gz
```

圧縮の使用はオプションですが、アーカイブの書き込み後の読み取り中に入出力の実行がより少ないため、通常圧縮した方がより高速です。詳細は、『[Oracle Solaris 11.3 での ZFS ファイルシステムの管理](#)』を参照してください。

6. ターゲットシステムでゾーンを構成します。

```
target# zonecfg -z my-zone -f /net/server/zonearchives/my-zone/my-zone.zonecfg
```

次のシステムメッセージが表示されます。

```
my-zone: No such zone configured
Use 'create' to begin configuring a new zone.
```

7. (オプション) 構成を表示します。

```
target# zonecfg:my-zone> info
zonename: my-zone
zonepath: /zones/my-zone
autoboot: false
pool:
net:
    address: 192.168.0.90
    physical: net0
```

8. 構成に必要な調整を加えます。

たとえば、ネットワーク物理デバイスがターゲットシステムで異なっていたり、構成の一部であるデバイスがターゲットシステムで異なる名前になっていたります。

```
target# zonecfg -z my-zone
zonecfg:my-zone> select net physical=net0
zonecfg:my-zone:net> set physical=net100
zonecfg:my-zone:net> end
```

9. 構成を確定して終了します。

```
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

10. 次のいずれかの方法を使用して、ゾーンをターゲットとなる移行先にインストールします。install サブコマンドを使用することをお勧めします。

- install が正常に完了できるように、必要な最小の更新を実行して、ゾーンをインストールします。

```
target# zoneadm -z my-zone install -p -a /net/server/zonearchives/my-zone/my-zone.zfs.gz
```

このリリースでは、接続が正常に完了できるように、必要な最小の更新を実行して、ゾーンの attach を行うこともできます。更新が許可されている場合は、`zoneadm attach` の実行中にパブリッシャーのカタログがリフレッシュされません。

```
target# zoneadm -z my-zone attach -u -a /net/server/zonearchives/my-zone/my-zone.zfs.gz
```

- **ゾーン内のすべてのパッケージを大域ゾーンと互換性のある最新バージョンに更新して、ゾーンをインストールします。**

```
target# zoneadm -z my-zone install -U -p -a /net/server/zonearchives/my-zone/my-zone.zfs.gz
```

このリリースでは、ゾーン内のすべてのパッケージを大域ゾーンと互換性のある最新バージョンに更新して、ゾーンの attach を行うこともできます。

```
target# zoneadm -z my-zone install -U -a /net/server/zonearchives/my-zone/my-zone.zfs.gz
```

- **ソフトウェアを更新せずに、ゾーンを新規ホストに接続します。**

```
target# zoneadm -z my-zone attach -a /net/server/zonearchives/my-zone/my-zone.zfs.gz
```

注記 - `install` サブコマンドを使用することをお勧めします。このリリースでは、ゾーン内のすべてのパッケージを大域ゾーンと互換性のある最新バージョンに更新して、ゾーンの attach を行うこともできます。

注意事項 ストレージオブジェクトに既存のパーティション、zpool、または UFS ファイルシステムが含まれる場合は、`install` が失敗し、エラーメッセージが表示されます。インストールを続行し、既存のデータをすべて上書きするには、`zoneadm install` に対して `-x force-zpool-create` オプションを使用します。

▼ ある ZFS プールから別のシステム内の別の ZFS プールにゾーンを移行する方法

次の手順は、ある ZFS プールから別のシステム内の別の ZFS プールにすべてのゾーンを移行する方法を示しています。

この手順で、ソース SysA 内のゾーンは、ターゲット SysB 内のプールに移動します。SysA 内の ZFS プールは `rpool/zones` であり、`/zones` でマウントされます。SysB 内のターゲットプールは `newpool/zones` です。移行されたゾーンの新しいゾーンパスは `/newpool/zones` です。

この手順ではまた、ターゲットプールがすでに存在することも前提にしています。

ZFS プールの作成の詳細は、『[Oracle Solaris 11.3 での ZFS ファイルシステムの管理](#)』の第 3 章、「[Oracle Solaris ZFS ストレージプールを作成および破棄する](#)」を参照してください。

始める前に 始める前に、次のことを確認する必要があります。

- ゾーンの移行を実行するためのすべての互換性要件が満たされていること。
- 両方のシステムでの鍵ベースの認証と root ログインを有効にするために SSH が構成されていること。SSH 認証の詳細は、『Oracle Solaris 11.3 での Secure Shell アクセスの管理』の「Secure Shell で使用する公開鍵と非公開鍵のペアを生成する方法」を参照してください。

1. 管理者になります。

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

2. rpool/zones を使用するすべてのゾーンを停止して切り離します。

この手順を、rpool/zones を使用しているすべてのゾーンに対して実行します。

```
sysA# zonename=zone1
sysA# zoneadm -z $zonename shutdown
sysA# zoneadm -z $zonename detach
sysA# zonecfg -z $zonename export -f /zones/$zonename.cfg
```

3. ソースプールのスナップショットを作成します。

```
sysA# zfs snapshot -r rpool/zones@send-to-sysB
```

4. ゾーンのストレージをシステム内のターゲットプールにレプリケートします。

```
sysA# zfs send -R rpool/zones@send-to-sysB | ssh sysB zfs receive -d newpool
```

5. ターゲットシステム上で、ゾーンを接続してブートします。この手順を、移行されたすべてのゾーンに対して実行します。

```
sysB# zonename=zone1
sysB# recvmountpoint=/newpool/zones
sysB# zonecfg -z $zonename -f /newpool/zones/$zonename.cfg
sysB# zonecfg -z $zonename 'set zonepath=/newpool/zones/{zonename}'
sysB# zoneadm -z $zonename attach -u
sysB# zoneadm -z $zonename boot
```

ここで、`{zonename}` は、`zonecfg(1M)` のマニュアルページの「トークン」のセクションに記載されているトークンです。

使用できないマシンからのゾーンの移行について

非大域ゾーンをホストするマシンが使用できなくなる場合があります。しかし、ゾーンが格納されている SAN などのストレージがまだ使用できる場合は、ゾーンを新しいホストに正常に移行できる可能性があります。ゾーンの `zonepath` を新しいホストに移動できます。SAN などのいくつかの場合には、`zonepath` データを実際には移動できないこともあります。SAN の場合は、`zonepath` が新規ホストに表示されるように、再構成が実行されるだけです。ゾーンが適切に切り離されなかったため、まず

zonecfg コマンドを使用して、新しいホスト上にゾーンを作成する必要があります。一度これを行なったあと、新しいホストでゾーンを接続してください。

このタスクの手順については、101 ページの「ZFS アーカイブを使用して別のホスト上の非大域ゾーンを変換したり、配備したりする方法」に記載されています。

大域ゾーンの非大域ゾーンへの変換

ゾーンは入れ子にならないため、P2V プロセスにより、変換されたシステムイメージ内の既存のゾーンはすべて、移行先ゾーンで使用できなくなります。統合アーカイブは、`archiveadm create -r -z` コマンドを使用してこの制限をバイパスします。詳細は、『Oracle Solaris 11.3 でのシステム復旧とクローン』の「復旧用アーカイブ」を参照してください。

Oracle Solaris システム (大域ゾーン) の solaris 非大域ゾーンへの変換について

既存の Oracle Solaris 11 システムを Oracle Solaris 11 システム上の solaris ブランドゾーンに直接移行できます。Oracle Solaris 11.2 以降を実行しているシステム上では、統合アーカイブの `archiveadm` コマンドを使用します。Oracle Solaris 11 または Oracle Solaris 11.1 を実行しているシステム上では、従来のアーカイブを使用します。

すべてのシステムについて、ソースシステム上で `zonep2vchk` コマンドを使用して、システムイメージの変換およびアーカイブの準備を行います。ターゲットシステム上で `zonecfg` および `zoneadm` コマンドを使用して、移行先ゾーン内のアーカイブを構成およびインストールします。

非大域ゾーンへの大域ゾーンの移行には、次の制約が適用されます。

- ターゲットシステム上の大域ゾーンで、元のソースホストと同じかそれ以降の Oracle Solaris 11 リリースが稼働している必要があります。
- ゾーンが適切に実行されるようにするには、必要なオペレーティングシステムパッケージの同じまたは以降のバージョンがターゲットシステムに存在する必要があります。サードパーティー製品のパッケージなど、ほかのパッケージは異なってもかまいません。

詳細

は、[zonep2vchk\(1M\)](#)、[archiveadm\(1M\)](#)、[zfs\(1M\)](#)、[zonecfg\(1M\)](#)、[zoneadm\(1M\)](#)、および [solaris\(5\)](#) のマニュアルページを参照してください。

▼ zonep2vchk を使用したソースシステムのスキャン

1. 管理者になります。

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

2. **zonep2vchk** ツールを **-b** オプションとともに使用して基本解析を実行し、**P2V** 移行によって影響を受ける可能性がある使用中の **Oracle Solaris** の機能がないかどうかを検査します。

```
source# zonep2vchk -b
```

3. **-s** オプションを指定して **zonep2vchk** ツールを実行し、アプリケーションファイルの静的解析を実行します。これは、**ELF** バイナリに、ゾーン内の操作に影響を与える可能性があるシステムコールとライブラリコールがないかどうかを検査します。

```
source# zonep2vchk -s /opt/myapp/bin,/opt/myapp/lib
```

4. **-r** オプションを指定して **zonep2vchk** ツールを実行し、ゾーン内で正常に実行できない可能性があるプロセスを探す実行時チェックを実行します。

```
source# zonep2vchk -r 2h
```

5. ソースシステム上で **-c** オプションを指定して **zonep2vchk** ツールを実行し、テンプレート **zonecfg** スクリプトを生成します (この手順では **s11-zone.config** という名前を付けます)。

```
source# zonep2vchk -c > /net/somehost/p2v/s11-zone.config
```

この構成には、ソースホストの物理リソースとネットワーク構成に基づくリソース制限とネットワーク構成が含まれます。

▼ ネットワークデバイス上にシステムイメージのアーカイブを作成する方法

大域ゾーン内のファイルシステムをアーカイブします。ソースシステムに非大域ゾーンがインストールされていないことを確認します。このセクションの例では、アーカイブの作成に **archiveadm** コマンドを使用します。

1. **管理者になります。**
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. **ネットワークストレージ上に大域ゾーンの回復用のアーカイブを作成します。**
この手順では、大域ゾーンの回復用のアーカイブの名前を **/net/somehost/p2v/s11-zone-config.uar** にします。

```
source# archiveadm create -r -z global /net/somehost/p2v/s11-zone-config.uar
```

参照 **archiveadm** で生成された統合アーカイブには、変換されたゾーンの開始点として機能するゾーン構成が含まれています。ターゲットシステムに最適な方法でゾーンを

構成するには、追加の構成が必要になることがあります。詳細は、『[Oracle Solaris 11.3 でのシステム復旧とクローン](#)』の「[復旧用アーカイブを作成する方法](#)」および [archiveadm\(1M\)](#)のマニュアルページを参照してください。

▼ ターゲットシステム上のゾーンを構成する方法

zonep2vchk ツールによって生成されたテンプレート zonecfg スクリプトは、ソースシステムの構成のうち、移行先のゾーン構成でサポートされる必要がある部分を定義します。ゾーンを完全に定義するには、ターゲットシステムに依存するその他の情報を手動で指定する必要があります。

この手順では、ゾーンの名前を s11-zone に、構成ファイルの名前を /net/somehost/p2v/s11-zone-config.uar にします。

1. 管理者になります。

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

2. アーカイブからゾーンを構成します。

```
target# zonecfg -z s11-zone
Use 'create' to begin configuring a new zone.
zonecfg:s11-zone> create -a /net/somehost/p2v/s11-zone-config.uar
zonecfg:s11-zone> info
```

3. ゾーンのストレージリソースを追加します。

```
zonecfg:s11-zone> set zonepath=/system/zones/{zonename}
zonecfg:s11-zone> add rootzpool
zonecfg:s11-zone:rootzpool> add storage iscsi://zfssa/luname.naa.600144F0DBF8AF19000052E820D60003
zonecfg:zonename:rootzpool> end
```

統合アーカイブを使用して配備されたゾーンを構成する場合は、ストレージリソースに ZOSS を使用することをお勧めします。[第13章「共有ストレージでの Oracle Solaris ゾーンの使用開始」](#)を参照してください。

4. ターゲットシステム上の大域ゾーンで次のコマンドを使用して、現在のリンク構成を表示します。

```
target# dladm show-link
target# dladm show-phys
target# ipadm show-addr
```

デフォルトでは、zonecfg スクリプトはソースシステム上で構成されたすべての物理ネットワークインタフェースに対して anet リソースを使用して排他的 IP ネットワーク構成を定義します。ターゲットシステムは、ゾーンのブート時に各 anet リソースに対して自動的に VNIC を作成します。VNIC を使用することで、複数のゾーンが同じ物理ネットワークインタフェースを共有できるようになります。anet リソースの下位リンク名は、zonecfg コマンドによって *change-me* に初期設定されます。この

フィールドは手動でターゲットシステム上のいずれかのデータリンクの名前に設定する必要があります。VNIC の下位リンクとして有効な任意のリンクを指定できます。

5. **ゾーン構成を確定して終了します。**

```
zonecfg:s11-zone> commit
zonecfg:s11-zone> exit
target#
```

▼ ターゲットシステムへのゾーンのインストール

この例では、インストール中に元のシステム構成は変更されません。

1. **管理者になります。**

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

2. **ソースシステム上で作成された統合アーカイブファイルを使用してゾーンをインストールします。**

```
target# zoneadm -z s11-zone install -a s11-zone-config.uar
```

ゾーンがインストールされている Oracle Solaris システムでの自動インストールおよびパッケージ

AI クライアントインストールの一環として、非大域ゾーンのインストールおよび構成を指定できます。このリリースでは、Image Packaging System (IPS) がサポートされています。この章では、ゾーンをインストールするときに、IPS パッケージを使用してオペレーティングシステムをインストールおよび保守する方法について説明します。

solaris10 ゾーンおよび native ゾーンで使用される SVR4 パッケージングおよびパッチについては、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の第 25 章「ゾーンがインストールされている Oracle Solaris システムでのパッケージについて (概要)」および第 26 章「ゾーンがインストールされている Oracle Solaris システムでのパッケージとパッチの追加および削除 (タスク)」を参照してください。これは、このガイドの Oracle Solaris 10 バージョンです。

Oracle Solaris リリースを実行しているシステムのイメージパッケージングシステムソフトウェア

グラフィカルツールおよびコマンド行ツールを使用すると、リポジトリからパッケージをダウンロードしてインストールできます。この章では、インストールされている非大域ゾーンにパッケージを追加するための情報を提供します。パッケージの削除に関する情報も含まれています。この章の内容は、既存の Oracle Solaris インストールおよびパッチに関するドキュメントの内容を補うものです。詳細は、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の第 3 章、「ソフトウェアパッケージのインストールおよび更新」を参照してください。

ゾーンのパッケージングの概要

solaris パッケージングリポジトリは、ゾーン環境の管理に使用されます。

pkg コマンドを使用してシステムを新しいバージョンの Oracle Solaris にアップグレードすると、ゾーンは自動的に更新されます。

イメージパッケージングシステム (IPS) (pkg(5) に記載) は、パッケージのインストール、アップグレード、削除などのソフトウェアのライフサイクル管理のために提供されるフレームワークです。IPS を使用すると、ソフトウェアパッケージの作成、パッケージリポジトリの作成と管理、および既存のパッケージリポジトリのミラー化を行うことができます。

Oracle Solaris オペレーティングシステムを初期インストールしたあと、イメージパッケージングシステムの CLI および GUI (パッケージマネージャー) クライアントを使用して、追加のソフトウェアアプリケーションをパッケージリポジトリからインストールできます。

システムにパッケージをインストールすると、IPS クライアントを使用してパッケージを検索、アップグレード、および管理できるようになります。また、IPS クライアントを使用して、システム全体を新しい Oracle Solaris リリースにアップグレードしたり、リポジトリの作成および管理や、既存のリポジトリのミラー化を行ったりすることもできます。

IPS がインストールされているシステムからインターネットにアクセスできる場合、クライアントは Oracle Solaris 11.3 パッケージリポジトリ (デフォルトの solaris パブリッシャー) <http://pkg.oracle.com/solaris/release/> にアクセスしてソフトウェアをインストールできます。

ゾーン管理者は、このドキュメントに記載されている制限の範囲内でパッケージツールを使用して、非大域ゾーンにインストールされたすべてのソフトウェアを管理できます。

ゾーンがインストールされている場合は、次の一般的な指針が適用されます。

- パッケージが大域ゾーンにインストールされている場合、非大域ゾーンでは大域ゾーンのシステムリポジトリサービスからパッケージをインストールすることができ、そのパッケージをインストールするためにネットワークを使用する必要はありません。そのパッケージが大域ゾーンにインストールされていない場合、ゾーンはゾーンプロキシサービスを使用してパブリッシャーにアクセスし、大域ゾーンを使用して、ネットワーク経由でパッケージをインストールする必要があります。
- 大域管理者または適切な承認を持つユーザーは、システムのすべてのゾーン内のソフトウェアを管理できます。
- 非大域ゾーンのルートファイルシステムは、Oracle Solaris パッケージツールを使用することで、大域ゾーンから管理できます。Oracle Solaris パッケージツールは、共通パッケージ (バンドル) 製品、スタンドアロン (別パッケージ) 製品、およびサードパーティー製品を管理するために、非大域ゾーン内でサポートされます。
- パッケージツールは、ゾーン対応の環境で動作します。このツールを使用すると、パッケージを非大域ゾーンにもインストールできます。

注記 - 特定のパッケージ操作が行われている間、ゾーンはこの種のほかの操作から一時的にロックされます。システムはまた、要求された操作について管理者に確認してから処理を続行することがあります。

パッケージとゾーンについて

solaris ブランドゾーンにインストールされているソフトウェア ([brands\(5\)](#) に記載) は、大域ゾーンにインストールされているソフトウェアと互換性がある必要があります。pkg コマンドを使用すると、この互換性が自動的に適用されます。大域ゾーンで pkg update コマンドを実行してソフトウェアを更新すると、ゾーンも更新されて、ゾーンと大域ゾーンの同期が維持されます。非大域ゾーンと大域ゾーンには、異なるソフトウェアをインストールできます。ゾーンで pkg コマンドを使用して、ゾーン内のソフトウェアを管理することもできます。

pkg update コマンド (FMRI の指定なし) を大域ゾーンで実行すると、pkg はシステムの大域ゾーンと非大域ゾーンのすべてのソフトウェアを更新します。

pkg install の試行 (予行演習とも呼ばれる) インストール機能を Oracle Solaris ゾーンで使用できます。試行インストールを実行するには、pkg-install -n コマンド構文を使用します。試行インストール中にシステムから拒否メッセージが生成された場合、問題の可能な解決策について、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[パッケージをインストールできない](#)」を参照してください。

ゾーンパッケージバリエーションを使用すると、パッケージ内の各種のコンポーネントが、大域ゾーン (global) または非大域ゾーン (nonglobal) のいずれかだけにインストールされるように明確にタグ付けされます。指定されるパッケージには、非大域ゾーンにインストールされないようにタグ付けされたファイルを含めることができます。

非大域ゾーンのインストール時に、大域ゾーンにインストールされた Oracle Solaris パッケージの一部だけが完全にレプリケートされます。たとえば、非大域ゾーン内では、Oracle Solaris カーネルを含む多数のパッケージは必要ありません。すべての非大域ゾーンは、大域ゾーンと同一のカーネルを暗黙的に共有します。

詳細は、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[非大域ゾーンの操作](#)」および『[Oracle Solaris 11.3 システムのインストール](#)』を参照してください。

注記 - 非大域ゾーンを含むシステム上で大域ゾーンを更新すると、ゾーンに関するパッケージダウンロード情報が 2 回表示されているように見える場合があります。しかし、パッケージがダウンロードされるのは一度だけです。

ゾーンがインストールされているシステムへのパッケージの追加について

Oracle Solaris 11 リリースで `pkg install` コマンドを使用します。

```
# pkg install package_name
```

大域ゾーンでの `pkg` の使用

パッケージを大域ゾーンのみを追加するには、大域ゾーンで `pkg install` コマンドを使用します。パッケージはほかのゾーンには伝達されません。

非大域ゾーンでの `pkg install` コマンドの使用

`pkg install` コマンドは、パッケージを非大域ゾーンだけに追加するために、ゾーン管理者が非大域ゾーンで使用します。指定された非大域ゾーン内にパッケージを追加するには、ゾーン管理者として `pkg install` コマンドを実行します。

パッケージの依存性は IPS で自動的に処理されます。

カスタム AI マニフェストを使用したゾーンへのパッケージの追加

AI マニフェストを改訂して、インストール時にゾーンにソフトウェアを追加する処理を自動化できます。指定したパッケージと、それらが依存するパッケージがインストールされます。パッケージのデフォルトのリストは AI マニフェストから取得されます。デフォルトの AI マニフェストは `/usr/share/auto_install/manifest/zone_default.xml` です。パッケージの特定方法および操作方法については、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』を参照してください。

例 5 マニフェストの改訂

次の手順では、`mercurial` と `vim` エディタの完全インストールを `my-zone` という名前の構成済みゾーンに追加します。(デフォルトでは、`solaris-small-server` の一部である最小限の `vim-core` のみがインストールされます。)

1. デフォルトの AI マニフェストをファイルの編集場所にコピーし、ファイルを書き込み可能にします。

```
# cp /usr/share/auto_install/manifest/zone_default.xml ~/my-zone-ai.xml
# chmod 644 ~/my-zone-ai.xml
```

2. ファイルを編集して、次のように mercurial および vim パッケージを software_data セクションに追加します。

```
<software_data action="install">
  <name>pkg:/group/system/solaris-small-server</name>
  <name>pkg:/developer/versioning/mercurial</name>
  <name>pkg:/editor/vim</name>
</software_data>
```

3. ゾーンをインストールします。

```
# zoneadm -z my-zone install -m ~/my-zone-ai.xml
```

システムには次のように表示されます。

```
A ZFS file system has been created for this zone.
Progress being logged to /var/log/zones/zoneadm.20111113T004303Z.my-zone.install
Image: Preparing at /zones/my-zone/root.
```

```
Install Log: /system/volatile/install.15496/install_log
AI Manifest: /tmp/manifest.xml.XfawpE
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Zonename: my-zone
Installation: Starting ...
```

```
Creating IPS image
Installing packages from:
solaris
origin: http://localhost:1008/
solaris/54453f3545de891d4daa841ddb3c844fe8804f55/
```

DOWNLOAD	PKGS	FILES	XFER (MB)
Completed	169/169	34047/34047	185.6/185.6

PHASE	ACTIONS
Install Phase	46498/46498

PHASE	ITEMS
Package State Update Phase	169/169
Image State Update Phase	2/2
Installation: Succeeded	
...	

ゾーン内でのパッケージの削除について

ゾーンがインストールされているシステムからパッケージを削除するには、`pkg uninstall` コマンドを使用します。

```
# pkg uninstall package_name
```

パッケージ情報の照会

ゾーンがインストールされているシステムでソフトウェアパッケージデータベースを照会するには、`pkg info` コマンドを使用します。

大域ゾーンでこのコマンドを使用すると、大域ゾーン内だけでソフトウェアパッケージデータベースを照会できます。非大域ゾーンでこのコマンドを使用すると、非大域ゾーン内だけでソフトウェアパッケージデータベースを照会できます。

インストール済みのゾーンがあるシステムでのプロキシ構成

『Oracle Solaris 11.3 ソフトウェアの追加と更新』の第5章、「インストールされるイメージの構成」で説明されているように、`--proxy` オプションを使用してイメージに永続的なプロキシを設定するようにしてください。永続的なイメージプロキシ構成を使用せず、`pkg` コマンドの実行時に常に `http_proxy` および `https_proxy` 環境変数を使用してリポジトリにアクセスする場合は、SMF の `system-repository` サービスプロパティを使用して、これらの同じプロキシを使用するように `system-repository` サービスも構成するようにしてください。[pkg\(1\)](#) のマニュアルページを参照してください。

大域ゾーンに構成されたりリポジトリへのアクセスは、`system-repository` サービスを使用して非大域ゾーンに提供されます。大域ゾーンの起点に関するプロキシの更新は、`system-repository` の構成に対して自動的に行われます。この方法を使用する場合、`system-repository` SMF サービスに対する変更は必要ありません。

大域ゾーンのパブリッシャーに構成されているプロキシをオーバーライドして、`system-repository` SMF サービスによって使用されるプロキシを構成することも可能です。`system-repository` プロキシは、`config/http_proxy` または `config/https_proxy` SMF プロパティを使用して設定できます。

詳細は、[pkg.sysrepo\(1M\)](#) のマニュアルページおよび『Oracle Solaris 11.3 ソフトウェアの追加と更新』を参照してください。

大域ゾーンでのプロキシの構成

大域ゾーンにはプロキシを直接構成することができ、大域ゾーンの起点に関するプロキシの更新は `system-repository` の構成に対して自動的に行われます。システムリポジトリサービスの変更は必要ありません。

例 6 大域ゾーンでのプロキシの構成

```
# pkg set-publisher --proxy http://www-proxy -g http://pkg-server pub
```

プロキシが 80 以外のポートで接続を受け入れる場合を除き、ポートを指定する必要はありません。

システム上にゾーンが存在する場合は、システムリポジトリサービスが再起動され、プロキシを使用して `pkg-server` へのアクセスが提供されます。

https_proxy および http_proxy を使用した system-repository プロキシのオーバーライド

イメージにプロキシを設定すること、および `system-repository` サービスプロキシのみを設定することをお勧めします。pkg コマンドを実行するときは、環境に `https_proxy` および `http_proxy` を設定するようにしてください。

このセクションの手順は、IPS パブリッシャーリポジトリに直接接続していない内部サブネット上の `system-repository` サービスにプロキシを設定するために使用します。この手順を使用すると、大域ゾーンで pkg コマンドによって構成されたすべてのプロキシがオーバーライドされます。非大域ゾーンは、HTTP を介して `system-repository` と通信します。`system-repository` は、大域ゾーンに構成されたそのリポジトリ用のプロトコルを使用してパブリッシャーにアクセスします。

この構成によって、solaris 非大域ゾーンから大域ゾーンに設定されているパブリッシャーにも接続できるようになります。solaris ゾーンへの再帰的 pkg 操作は成功します。

例 7 https_proxy および http_proxy を使用した大域ゾーンのプロキシのオーバーライド

たとえば、solaris 非大域ゾーンを実行しているシステム上のソフトウェアが IPS によって管理されていて、http と https の両方の URL にアクセスするためにプロキシサーバー `http_proxy=http://192.168.0.0` を使用する必要があるとします。次の手順は、`http_proxy` および `https_proxy` 環境変数と SMF サービスプロパティを使用して、大域ゾーンと非大域ゾーンから IPS リポジトリにアクセスできるようにする方法を示しています。

これらの変数は、ユーザーが非大域ゾーンから `pkg` コマンドを実行してシステムパブリッシャーの URI (Universal Resource Identifier) に接続しないかぎり、起点に設定されたプロキシ構成をオーバーライドします。この場合、コマンドは `system-repository` を経由します。

解決可能なホスト名を使用することもできます。

1. 次の行を入力して、大域ゾーンの `shell` にプロキシを設定します。

```
# export http_proxy=http://192.168.0.0
# export https_proxy=http://192.168.0.0
```

プロキシを設定すると、`pkg` コマンドがプロキシサーバー経由でパブリッシャーに到達できるようになります。これは、`https` または `http` URL を使用し、大域ゾーンの `system-repository` を経由しない `pkg` 操作に影響を与えます。

2. システム上の `solaris` ゾーンが大域ゾーンから直接アクセスできる構成済みのシステムパブリッシャーを使用できるようにするには、次のコマンドを実行します。

```
# svccfg -s system-repository:default setprop config/http_proxy = http://192.168.0.0
# svccfg -s system-repository:default setprop config/https_proxy = http://192.168.0.0
```

3. ライブ SMF リポジトリ内で変更を有効にするには、次を実行します。

```
# svcadm refresh system-repository
```

4. 設定が機能していることを確認するには、次を実行します。

```
# svcprop -p config/http_proxy system-repository
# svcprop -p config/https_proxy system-repository
```

`pkg` コマンドの詳細は、[pkg\(1\)](#) のマニュアルページを参照してください。

ゾーンの並列更新

ゾーンは、順次ではなく並列的に更新されるように構成できます。並列更新により、システム上のすべてのゾーンを更新するのに必要な時間が大幅に短縮されます。詳細と構成例については、『[Oracle Solaris 11.3 ソフトウェアの追加と更新](#)』の「[複数の非大域ゾーンの同時更新](#)」を参照してください。

ゾーンの状態がパッケージの操作に与える影響

さまざまな状態の非大域ゾーンが存在しているシステムでパッケージコマンドを使用した結果を、次の表に示します。

ゾーンの状態	パッケージ操作への影響
構成済み	パッケージツールを実行できます。ソフトウェアはまだインストールされていません。
不完全	ゾーン上で <code>zoneadm</code> が動作している場合は、パッケージツールを使用しないようにしてください。ゾーン上で <code>zoneadm</code> プロセスが動作していない場合は、パッケージ操作を実行しても安全ですが、このゾーン内のソフトウェアは変更されず、ゾーン内のソフトウェアは依存性の解決に影響を与えません。
使用不可	ゾーン内のソフトウェアイメージにアクセスできません。ソフトウェアイメージは変更されず、依存性の解決にも影響を与えません。
インストール済み	パッケージツールを実行できます。 <code>zoneadm -z zonename install</code> が完了した直後にも、ゾーンはインストール済み状態に移行されません。
準備完了	パッケージツールを実行できます。
実行中	パッケージツールを実行できます。

非大域ゾーンのストレージにアクセスできない場合、または非大域ゾーンのイメージ (`pkg(5)` に記載) が大域ゾーンのイメージと同期していない場合、そのゾーンは使用不可の状態に移行します。この状態移行は、非大域ゾーンに影響を与えている問題によって大域ゾーンのパッケージ操作がブロックされないようにするために行われます。

ゾーンのストレージが一時的に使用できなくなり、インストール済みソフトウェアのバージョンを変更するパッケージ操作が行われたときは、ストレージの問題を修正したあとで、更新を可能にする `solaris` ブランドのいずれかの `attach` オプションを使用してゾーンを接続する必要がある場合があります。たとえば、大域ゾーンと使用不可の状態になっている非大域ゾーンの間で重要なソフトウェアのバージョンを同期するには、`zoneadm -z zonename attach -u` を使用する必要がある場合があります。

Oracle Solaris ゾーンの管理について

この章では、次の一般的なゾーン管理について説明します。

- 120 ページの「大域ゾーンの可視性とアクセス」
- 120 ページの「ゾーン内でのプロセス ID の可視性」
- 121 ページの「ゾーン内のシステム監視機能」
- 121 ページの「zonestat ユーティリティを使用したアクティブなゾーンの統計情報の報告」
- 122 ページの「fsstat ユーティリティを使用した非大域ゾーンのモニタリング」
- 122 ページの「非大域ゾーンのノード名」
- 123 ページの「ファイルシステムと非大域ゾーン」
- 131 ページの「共有 IP 非大域ゾーンにおけるネットワーク」
- 134 ページの「排他的 IP 非大域ゾーンにおけるネットワーク」
- 136 ページの「非大域ゾーンでのデバイスの使用」
- 139 ページの「非大域ゾーンでのアプリケーションの実行」
- 139 ページの「非大域ゾーンで使用されるリソース制御」
- 140 ページの「ゾーンがインストールされているシステムでの公平配分スケジューラ」
- 140 ページの「ゾーンがインストールされているシステムでの拡張アカウントリング」
- 141 ページの「非大域ゾーン内の特権」
- 145 ページの「ゾーン内での IP セキュリティアーキテクチャーの使用」
- 146 ページの「ゾーン内での Oracle Solaris 監査の使用」
- 146 ページの「ゾーン内のコアファイル」
- 147 ページの「非大域ゾーン内での DTrace の実行」
- 148 ページの「ゾーンがインストールされている Oracle Solaris システムのバックアップについて」
- 149 ページの「非大域ゾーン内でバックアップするデータの決定」
- 151 ページの「ゾーンがインストールされているシステムで使用するコマンド」

solaris10 ブランドゾーンについては、『Oracle Solaris 10 ゾーンの実行と使用』を参照してください。

大域ゾーンの可視性とアクセス

大域ゾーンは、システムのデフォルトゾーンとしても、システム規模の管理制御用ゾーンとしても機能します。この二重の役割と関係のある管理上の問題が存在します。ゾーン内部のアプリケーションはほかのゾーン内のプロセスおよびほかのシステムオブジェクトにアクセスするため、管理操作の効果が予想したものよりも大きい場合があります。たとえば、サービスの停止スクリプトでは、プロセスを終了させるためのシグナルの送信にしばしば `pkill` が使用されます。大域ゾーンからこの種のスクリプトを実行すると、システム内の該当するすべてのプロセスが、ゾーンに関係なくシグナルを送信します。

多くの場合、システム規模の範囲が必要になります。たとえば、システム規模のリソース使用状況をモニターする場合、システム全体のプロセス統計情報を表示する必要があります。大域ゾーンのアクティビティだけのビューには、システムリソースの一部または全体を共有可能なシステム内のほかのゾーンからの関連情報が欠落しています。この種のビューは、CPU などのシステムリソースがリソース管理機能を使用して厳密に区分されていない場合、特に重要です。

このため、大域ゾーン内のプロセスから、非大域ゾーン内のプロセスおよびほかのオブジェクトを監視できます。これにより、この種のプロセスがシステム規模の監視機能を備えることが可能になります。ほかのゾーン内のプロセスを制御したりシグナルを送信したりする機能は、`PRIV_PROC_ZONE` 特権により制限されません。この特権は、特権のないプロセスに設定された制限をオーバーライドできるため、`PRIV_PROC_OWNER` に類似しています。この場合の制限は、大域ゾーン内の特権のないプロセスはほかのゾーン内のプロセスにシグナルを送信したり制御したりすることはできない、というものです。これは、プロセスのユーザー ID が一致するか、動作しているプロセスが `PRIV_PROC_OWNER` 特権を保持している場合でも適用されません。`PRIV_PROC_ZONE` 特権を、そうでなければ特権の付与されたプロセスから削除して、大域ゾーンへの操作に制限できます。

`zoneidlist` を使用した照合プロセスの詳細は、[pgrep\(1\)](#) および [pkill\(1\)](#) のマニュアルページを参照してください。

ゾーン内でのプロセス ID の可視性

同一ゾーン内のプロセスだけが、`kill` や `priocntl` コマンドなどの、プロセス ID を指定するシステムコールインタフェースを介して表示されます。詳細は、[kill\(1\)](#) および [priocntl\(1\)](#) のマニュアルページを参照してください。

ゾーン内のシステム監視機能

大域ゾーンで `ps` コマンドを実行する場合、ユーザーおよびグループ名は大域ゾーンのネームサービスを使用して解決されます。システムの非大域ゾーンで実行しているプロセスは、大域ゾーンのネームサービスに一致したユーザーおよびグループ名を表示します。これらの大域ゾーン名は、非大域ゾーンの名前サービスで構成された名前とは異なる可能性があります。

`ps` コマンドに、次の変更が加えられました。

- 出力形式の指定には `-o` オプションを使用します。このオプションを使用すると、プロセスのゾーン ID またはプロセスを実行中のゾーンの名前を出力できます。
- 指定されたゾーン内のプロセスだけをリスト表示するには、`-z zonelist` オプションを使用します。ゾーンの指定には、ゾーン名またはゾーン ID を使用できます。このオプションは、大域ゾーン内でコマンドを実行する場合にのみ有効です。
- プロセスに関連するゾーンの名前を出力するには `-Z` オプションを使用します。列見出し `ZONE` の下に名前が出力されます。

詳細は、[ps\(1\)](#) のマニュアルページを参照してください。

`-z zonename` オプションが、次の Oracle Solaris ユーティリティに追加されました。このオプションを使用して情報をフィルタ処理し、指定したゾーンだけを含めることができます。

- `ipcs` ([ipcs\(1\)](#)) のマニュアルページを参照)
- `pgrep` ([pgrep\(1\)](#)) のマニュアルページを参照)
- `ptree` ([proc\(1\)](#)) のマニュアルページを参照)
- `prstat` ([prstat\(1M\)](#)) のマニュアルページを参照)

コマンドに加えられた変更の全一覧については、[表5](#)を参照してください。

zonestat ユーティリティを使用したアクティブなゾーンの統計情報の報告

`zonestat` ユーティリティを使用する場合は、[zonestat\(1\)](#) のマニュアルページおよび[159 ページ](#)の「非大域ゾーンでの `zonestat` ユーティリティの使用」を参照してください。

`zonestat` ユーティリティは、現在実行中のゾーンの CPU、メモリー、およびリソース制御の使用効率について報告します。`zonestat` ユーティリティは、指定された間隔で一連のレポートを出力します。必要に応じて、このユーティリティは 1 つ以上のサマリーレポートを出力できます。

zonestat ユーティリティは、排他的 IP ゾーン内のネットワーク帯域幅の使用効率についても報告を生成します。排他的 IP ゾーンには、独自の IP に関連付けられた状態と 1 つ以上の専用のデータリンクが保持されます。

非大域ゾーンから実行すると、そのゾーンから参照可能なプロセッサセットのみが報告されます。非大域ゾーンの出力には、すべてのメモリーリソースおよび制限リソースが含まれます。

非大域ゾーンで zonestat サービスを使用するには、大域ゾーンの zonestat サービスがオンラインである必要があります。各非大域ゾーンの zonestat サービスは、大域ゾーンの zonestat サービスからシステム構成および使用効率データを読み取ります。

zonestatd システムデーモンは、システムブート時に開始されます。このデーモンは、ゾーンによるシステムリソースの使用をモニターするほか、psrset プロセッサセット、プールプロセッサセット、リソース制御設定などのゾーンおよびシステム構成情報をモニターします。構成できるコンポーネントはありません。

fsstat ユーティリティを使用した非大域ゾーンのモニタリング

fsstat ユーティリティは、ゾーンごとの kstats (アグリゲーションを含む) を収集して出力します。デフォルトでは、このユーティリティは実行中のすべてのゾーンの集計を報告します。fstype ごとの kstat がそれぞれのゾーン用に生成されます。大域ゾーンの kstat は、その排他的なアクティビティを報告します。大域ゾーンでは、システム上のすべてのゾーンの kstat を表示できます。非大域ゾーンでは、ユーティリティが実行されているゾーンに関連付けられた kstats のみが表示されます。非大域ゾーンでは、ほかのゾーンのファイルシステムアクティビティをモニターできません。

詳細は、[fsstat\(1M\)](#) のマニュアルページおよび [162 ページの「すべてのゾーンのゾーン別の fstype 統計情報の報告」](#) を参照してください。

非大域ゾーンのノード名

ノード名は、システム名のローカルソースです。ノード名はゾーン名などのように一意である必要があります。ノード名は、ゾーン管理者が設定できます。

```
# hostname myhostname
```

ホスト名を表示するには、hostname と入力します。

```
# hostname  
...
```

myhostname

ゾーンでの NFS サーバーの実行

ゾーン内に NFS 共有を作成するには、そのゾーン内に NFS サーバーパッケージ `svc:/network/nfs/server:default` をインストールする必要があります。

ゾーン内での NFS 共有を防止するため、ゾーン構成で `sys_share` 特権を禁止できません。表1を参照してください。

次の制限事項があります。

- ゾーンをまたぐ LOFS マウントはゾーンから共有できません。
- ゾーン内にマウントされたファイルシステムは、大域ゾーンから共有できません。
- リモートダイレクトメモリアccess (RDMA) を使用した NFS は、ゾーンでサポートされません。
- Oracle Sun Cluster HA for NFS (HANFS) のフェイルオーバーは、ゾーンでサポートされません。

『Oracle Solaris 11 ネットワークサービスの紹介』を参照してください。

ファイルシステムと非大域ゾーン

このセクションでは、ゾーンがインストールされている Oracle Solaris システムでファイルシステムを使用する場合の問題について説明します。各ゾーンは、ゾーンの `root` と呼ばれるディレクトリをルートとする、ファイルシステム階層の独自セクションを保持します。ゾーン内のプロセスは、ゾーンルート以下の階層部分内のファイルだけにアクセスできます。ゾーン内で `chroot` ユーティリティーを使用できますが、プロセスをゾーン内のルートパスに制限する場合だけです。`chroot` の詳細については、[chroot\(1M\)](#) のマニュアルページを参照してください。

-o nosuid オプション

`mount` ユーティリティーで `-o nosuid` オプションを指定する場合、次の機能を利用できます。

- `nosetuid` オプションを使用してマウントされたファイルシステム上の `setuid` バイナリに基づくプロセスは、`setuid` バイナリの権限では動作しません。プロセスは、バイナリを実行するユーザーの権限で動作します。

たとえば、ユーザーが `root` の所有する `setuid` バイナリを実行する場合、プロセスはそのユーザーの権限で動作します。

- ファイルシステム内のデバイス特殊エントリを開くことはできません。この動作は、`nodevices` オプションを指定する場合と同じです。

`mount(1M)` のマニュアルページに記載されているように、このファイルシステム固有のオプションは、`mount` ユーティリティでマウントできるすべての Oracle Solaris ファイルシステムで使用できます。これらのファイルシステムの一覧については、このガイドの [124 ページの「ゾーン内でのファイルシステムのマウント」](#) を参照してください。マウント機能についても説明します。`-o nosuid` オプションの詳細は、『[Managing Network File Systems in Oracle Solaris 11.2](#)』の第7章、「[Accessing Network File Systems](#)」を参照してください。

ゾーン内でのファイルシステムのマウント

ファイルシステムをゾーン内部でマウントする場合、`nodevices` オプションが適用されます。たとえば、ゾーンに、UFS ファイルシステムに対応するブロック型デバイス (`/dev/dsk/c0t0d0s7`) および raw デバイス (`/dev/rdsk/c0t0d0s7`) へのアクセスが許可される場合、ゾーン内部からマウントを行うと、ファイルシステムのマウントで自動的に `nodevices` オプションが適用されます。この規則は、`zonecfg` 構成を使用して指定されたマウントには適用されません。

次の表で、非大域ゾーン内でファイルシステムをマウントする場合のオプションを説明します。これらの補助的なマウント方法の実行手順については、[22 ページの「ゾーンを構成、検証、および確定する」](#) および [165 ページの「稼働中の非大域ゾーン内でファイルシステムをマウントする」](#) を参照してください。

`/usr/lib/fstype/mount` 内にマウントバイナリが存在する場合、表に含まれない任意のファイルシステムタイプを構成内で指定できます。

非大域ゾーンの内部から HSFS と NFS 以外のファイルシステムタイプをマウントするには、`zonecfg fs-allowed` プロパティを使用して、構成にファイルシステムタイプも追加します。

デフォルト以外のファイルシステムマウントを許可すると、ゾーン管理者によってシステムが危険にさらされることがあります。

ファイルシステム	非大域ゾーン内のマウントオプション
AutoFS	マウントに <code>zonecfg</code> を使用できません。ゾーン内部からマウントすることは可能です。
CacheFS	非大域ゾーン内では使用できません。
FDfs	<code>zonecfg</code> を使用してマウントできます。ゾーン内部からマウントできます。
HSFS	<code>zonecfg</code> を使用してマウントできます。ゾーン内部からマウントできます。

ファイルシステム	非大域ゾーン内のマウントオプション
LOFS	zonecfg を使用してマウントできます。ゾーン内からマウントできます。
MNTFS	マウントに zonecfg を使用できません。ゾーン内部からマウントすることは可能です。
NFS	マウントに zonecfg を使用できません。ゾーン内で現在サポートされているバージョンである V2、V3、および V4 を、ゾーン内からマウントできます。
PCFS	zonecfg を使用してマウントできます。ゾーン内からマウントできます。
PROCFS	マウントに zonecfg を使用できません。ゾーン内部からマウントすることは可能です。
TMPFS	zonecfg を使用してマウントできます。ゾーン内からマウントできます。
UDFS	zonecfg を使用してマウントできます。ゾーン内からマウントできます。
UFS	<p>zonecfg を使用してマウントできます。ゾーン内からマウントできます。</p> <p>注記 - quota コマンド (quota(1M) を参照) は、zonecfg add fs リソースを使用して追加された UFS ファイルシステムの割り当て情報の取得に使用することはできません。</p> <p>add fs を使用する場合は、system/file-system/ufs パッケージを大域ゾーンにインストールする必要があります。zonecfg コマンドを使用して非大域ゾーンで UFS ファイルシステムを使用する場合は、インストール後にこのパッケージをゾーンにインストールするか、AI マニフェストスクリプトを使用してインストールする必要があります。</p> <p>次を 1 行で入力します。</p> <pre>global# pkg -R /tank/zones/my-zone/root \ install system/file-system/ufs</pre>
VxFS	zonecfg を使用してマウントできます。ゾーン内からマウントできます。
ZFS	zonecfg dataset および fs リソースタイプを使用してマウントできます。

詳細は、[22 ページの「ゾーンの構成方法」](#)、[165 ページの「稼働中の非大域ゾーン内でファイルシステムをマウントする」](#)、および [mount\(1M\)](#) のマニュアルページを参照してください。

ゾーン内でのファイルシステムのアンマウント

ファイルシステムをアンマウントできるかどうかは、どの管理者がそのファイルシステムを最初にマウントしたかによって決まります。zonecfg コマンドを使用してゾーンを構成するときにファイルシステムが指定されている場合は、大域ゾーンがそのマウントの所有者となるので、非大域ゾーンの管理者はそのファイルシステムをアンマウントすることはできません。非大域ゾーンの /etc/vfstab ファイルにマウントを指定するなどの方法でファイルシステムが非大域ゾーンからマウントされている場合には、その非大域ゾーンの管理者はそのファイルシステムをアンマウントできます。

セキュリティの制限およびファイルシステムの動作

ゾーン内部から特定のファイルシステムをマウントする場合、適用されるセキュリティの制限が存在します。ほかのファイルシステムは、ゾーン内でマウントされたときに特有の動作を行います。変更されたファイルシステムの一覧を、次に示します。

AutoFS

autofs は、自動的に適切なファイルシステムをマウントするためのクライアント側のサービスです。クライアントが現在マウントされていないファイルシステムにアクセスしようとする時、AutoFS ファイルシステムはその要求に介入し、automountd を呼び出して要求されたディレクトリをマウントします。ゾーン内で確立された AutoFS マウントは、そのゾーンだけで有効です。大域ゾーンを含むほかのゾーンからそのマウントにアクセスすることはできません。ゾーンが停止またはリブートすると、マウントは削除されます。AutoFS の詳細は、『[Oracle Solaris 11.3 でのネットワークファイルシステムの管理](#)』の「[autofs のしくみ](#)」を参照してください。

各ゾーンは、automountd の独自コピーを実行します。自動マップおよびタイムアウトは、ゾーン管理者により制御されます。大域ゾーンから非大域ゾーンの AutoFS マウントポイントを横断的に使用して、別のゾーン内でマウントを始動させることはできません。

別のマウントが始動する際、カーネル内で特定の AutoFS マウントが作成されます。この種のマウントは、一括してマウントまたはアンマウントする必要があるため、通常の umount インタフェースを使用して削除することはできません。この機能は、ゾーンの停止処理用であることに注意してください。

MNTFS

MNTFS は、ローカルシステムのマウント済みファイルシステムのテーブルに読み取り専用アクセスを提供する仮想ファイルシステムです。非大域ゾーン内から mnttab を使用して表示可能な

および修正します。fsck が失敗した場合、zoneadmd を使用してゾーンを準備完了状態にすることはできません。raw により指定されるパスを、相対パスにすることはできません。

/usr/lib/fs/fstype/fsck 内で fsck バイナリを提供しないファイルシステムで、デバイスを fsck に指定することは誤りです。また、そのファイルシステムに fsck バイナリが存在する場合に、デバイスを fsck に指定しないことも誤りです。

詳細は、41 ページの「zoneadmd デーモン」および fsck(1M) コマンドを参照してください。

ZFS

デフォルトのデータセット (『Oracle Solaris ゾーン構成リソース』の「ゾーンでマウントされるファイルシステム」に記載)に加えて、zonecfg コマンドに add dataset リソースを付けて実行することにより、ZFS データセットを非大域ゾーンに追加できます。データセットは非大域ゾーンでマウントされ可視になり、大域ゾーンでも可視になります。ゾーン管理者は、そのデータセット内のファイルシステムの作成と破棄、およびデータセットのプロパティの変更を行うことができます。

zfs の zoned 属性は、データセットが非大域ゾーンに追加されたかどうかを示します。

```
# zfs get zoned tank/sales
NAME          PROPERTY  VALUE   SOURCE
tank/sales    zoned     on      local
```

データセットリソースを使用して非大域ゾーンに委任された各データセットには、別名が付けられます。データセットのレイアウトはゾーン内では表示されません。別名が付けられた各データセットは、プールのようにゾーンに表示されます。データセットのデフォルトの別名は、データセット名の最後のコンポーネントです。たとえば、委任されたデータセット tank/sales にデフォルトの別名が使用された場合、ゾーンには sales という仮想 ZFS プールが表示されます。別名が別の値になるようにカスタマイズするには、データセットリソース内の別名プロパティを設定します。

各非大域ゾーンの zonepath データセット内には、rpool というデータセットが存在します。すべての非大域ゾーンで、ゾーンのこの rpool データセットは rpool として別名が付けられます。

```
my-zone# zfs list -o name,zoned,mounted,mountpoint
NAME          ZONED  MOUNTED  MOUNTPOINT
rpool         on     no       /rpool
rpool/ROOT    on     no       legacy
rpool/ROOT/solaris on     yes      /
rpool/export  on     no       /export
rpool/export/home on     no       /export/home
```

データセットの別名は、ZFS プールと同じ名前制限の対象となります。この制限については、[zpool\(1M\)](#) のマニュアルページに記載されています。

大域ゾーンのデータセットを共有する場合は、`zonecfg` コマンドとともに `add fs` サブコマンドを使用して、LOFS マウントした ZFS ファイルシステムを追加できます。データセットのプロパティの設定および制御は、大域管理者または適切な承認を付与されたユーザーが担当します。

ZFS の詳細は、『[Oracle Solaris 11.2 での ZFS ファイルシステムの管理](#)』の第 9 章、「[Oracle Solaris ZFS の高度なトピック](#)」を参照してください。

NFS クライアントとして機能する非大域ゾーン

ゾーンは、NFS クライアントとしても機能できます。バージョン 2、バージョン 3、およびバージョン 4 プロトコルがサポートされます。これらの NFS バージョンについては、『[Oracle Solaris 11.3 でのネットワークファイルシステムの管理](#)』の「[NFS サービスの機能](#)」を参照してください。

デフォルトのバージョンは、NFS バージョン 4 です。次のいずれかの方法を使用して、クライアント上でほかの NFS バージョンを有効にできます。

- `sharectl(1M)` を使用すると、プロパティを設定できます。 `NFS_CLIENT_VERSMAX=number` を設定することで、指定したバージョンをゾーンのデフォルトとして使用できます。『[Oracle Solaris 11.3 でのネットワークファイルシステムの管理](#)』の「[NFS サービスの設定](#)」を参照してください。
- バージョンマウントを手動で作成できます。この方法は、`sharectl` 設定をオーバーライドします。『[Oracle Solaris 11.3 でのネットワークファイルシステムの管理](#)』の「[NFS サービスの設定](#)」を参照してください。

ゾーン内での `mknod` の使用禁止

`mknod(1M)` のマニュアルページに記載されている `mknod` コマンドを使用して、非大域ゾーンで特殊ファイルを作成することはできません。

ファイルシステムの行き来

ゾーンのファイルシステム名前空間は、大域ゾーンからアクセス可能な名前空間の一部です。大域ゾーン内の特権のないプロセスが、非大域ゾーンのファイルシステム階層を行き来することはできません。これには、次のような理由があります。

- root だけが、ゾーンルートの親ディレクトリを所有、読み込み可能、書き込み可能、および実行可能に指定する
- /proc によりエクスポートされたディレクトリへのアクセスが制限される

別のゾーン用にマウントされた AutoFS ノードへのアクセスを試みても、失敗します。大域管理者は、その子孫がほかのゾーンに含まれる自動マップを保持してはいけません。

大域ゾーンから非大域ゾーンにアクセスする際の制限

非大域ゾーンのインストール後に、システムのバックアップユーティリティー以外のコマンドを使用して、大域ゾーンからそのゾーンに直接アクセスしてはいけません。また、非大域ゾーンを未知の環境に公開した後は、そのゾーンが安全であると考えすることはできません。たとえば、公開されたネットワーク上に配置されたゾーンについて考えてみましょう。この場合、ゾーンのセキュリティが低下し、ファイルシステムの内容が変更される可能性があります。セキュリティが低下する可能性がある場合、大域管理者はゾーンを信頼できないものとして処理する必要があります。

次の両方の条件が当てはまる場合、`-R` オプションまたは `-b` オプション (またはこれと同等なオプション) を使って代替ルートを指定可能なコマンドを使用してはいけません。

- コマンドが大域ゾーン内で実行される。
- 代替ルートが非大域ゾーン内のいずれかのパスを参照している。現在動作中のシステムの大域ゾーンからの相対パスの場合や、代替ルート内の大域ゾーンからの相対パスの場合を含む。

たとえば、`pkgadd` ユーティリティーに `-R root_path` オプションを指定して、非大域ゾーンのルートパスを保持する大域ゾーンから実行する場合は、この条件に該当します。

次に、`-R` と代替ルートパスを使用するコマンド、プログラム、およびユーティリティーを一覧表示します。

- `auditreduce`
- `bart`
- `installf`
- `localeadm`
- `makeuuid`
- `metaroot`
- `pkg`
- `prodreg`

- removef
- routeadm
- showrev
- syseventadm

次に、`-b` と代替ルートパスを使用するコマンドおよびプログラムを一覧表示します。

- add_drv
- pprosetup
- rem_drv
- roleadd
- update_drv
- useradd

共有 IP 非大域ゾーンにおけるネットワーク

Oracle Solaris システムにインストールしたゾーンはネットワーク経由で互いに通信できます。ゾーンは別個の結合または接続を保持します。また、独自のサーバーデーモンを実行できます。これらのデーモンは、同一のポート番号で競合することなく待機できます。IP スタックは、着信接続の IP アドレスを考慮に入れることで競合を解決します。IP アドレスにより、ゾーンが識別されます。

Oracle Solaris では、特定のシステムのネットワーク構成は、Oracle Solaris インストール中に自動的に有効にされるか、システム管理者が手動で有効にするアクティブなネットワーク構成プロファイル (NCP) で管理されます。システム上で一度にアクティブになることができる NCP は 1 つだけです。

共有 IP タイプのゾーンを使用するには、大域ゾーンでの有効な NCP が DefaultFixed NCP である必要があります。システムでどの NCP が現在アクティブになっているかを調べるには、次のコマンドを入力します。

```
# netadm list
TYPE  PROFILE          STATE
ncp   DefaultFixed    online
ncp   Automatic        disabled
loc   Automatic        offline
loc   NoNet            offline
loc   DefaultFixed     online
```

online 状態の NCP は、システムで有効またはアクティブなネットワーク構成プロファイルです。DefaultFixed がオフラインの場合、次のコマンドでプロファイルを有効にします。

```
# netadm enable DefaultFixed
# svcprop -p netcfg/active_ncp svc:/network/physical:default
```

DefaultFixed

共有 IP ゾーンの区分化

共有 IP はデフォルトではありませんが、このタイプはサポートされます。

ゾーンをサポートするシステム内の IP スタックは、ゾーン間のネットワークトラフィックの分離を実装します。IP トラフィックを受け取るアプリケーションは、同じゾーンに送信されたトラフィックの受信だけを実行できます。

システム上の各論理インタフェースは、特定のゾーンに所属します。デフォルトは、大域ゾーンです。zonecfg ユーティリティ経由でゾーンに割り当てられた論理ネットワークインタフェースは、ネットワーク経由での通信に使用されます。各ストリームおよび接続は、それを開いたプロセスのゾーンに所属します。

上位層ストリームと論理インタフェース間の結合は、制限されます。ストリームが確立できるのは、同一ゾーン内の論理インタフェースへの結合だけです。同様に、論理インタフェースからのパケットを渡すことができるのは、論理インタフェースと同じゾーン内の上位層ストリームに対してだけです。

各ゾーンは、独自のバインドセットを保持します。アドレスが使用中であるため、各ゾーンは、同一のポート番号で待機する同じアプリケーションを、バインドが失敗することなく稼働可能です。各ゾーンは、次のような各種のネットワークサービスの固有のバージョンを実行できます。

- 完全な構成ファイルを保持するインターネットサービスデーモン ([inetd\(1M\)](#) のマニュアルページを参照してください)
- sendmail ([sendmail\(1M\)](#) のマニュアルページを参照)
- apache

大域ゾーン以外のゾーンは、ネットワークへのアクセスが制限されています。標準の TCP および UDP ソケットインタフェースが利用可能ですが、SOCK_RAW ソケットインタフェースは ICMP (Internet Control Message Protocol) に制限されています。ICMP は、ネットワークのエラー状況を検出および報告したり、ping コマンドを使用するのに必要です。

共有 IP ネットワークインタフェース

ネットワーク接続を必要とする非大域ゾーンには、それぞれ 1 つ以上の専用 IP アドレスがあります。これらのアドレスは、ゾーン内に配置可能な論理ネットワークインタフェースに関連付けられています。zonecfg により構成されるゾーンネットワークインタフェースは、ブート時に自動的に設定されてゾーン内に配置されます。ipadm コ

マンドを使用すると、ゾーンの稼働中に論理インタフェースを追加または削除できます。インタフェース構成およびネットワーク経路を変更できるのは、大域管理者または適切な承認を付与されたユーザーのみです。

非大域ゾーン内では、そのゾーンのインタフェースのみが `ipadm` コマンドで表示されます。

詳細は、[ipadm\(1M\)](#) および [if_tcp\(7P\)](#) のマニュアルページを参照してください。

同一マシン上の共有 IP ゾーン間の IP トラフィック

指定された IP 宛先への使用可能な経路がルーティングテーブルにある場合、共有 IP ゾーンはその宛先に到達できます。ルーティングテーブルを表示するには、ゾーン内から `netstat` コマンドに `-r` オプションを付けて実行します。IP 転送ルールは、別のゾーンまたは別のシステムの IP 宛先でも同じです。

共有 IP ゾーンでの Oracle Solaris IP フィルタ

Oracle Solaris IP フィルタは、ステートフルパケットフィルタリングとネットワークアドレス変換 (NAT) を行います。ステートフルパケットフィルタは、アクティブな接続の状態をモニターし、取得した情報を使用して、ファイアウォールの通過を許可するネットワークパケットを決定することができます。Oracle Solaris IP フィルタには、ステートレスパケットフィルタリングと、アドレスプールの作成および管理を行う機能もあります。追加情報については、『[Oracle Solaris 11.3 でのネットワークのセキュリティ保護](#)』の第 6 章、「[Oracle Solaris での IP フィルタファイアウォール](#)」を参照してください。

『[Oracle Solaris 11.3 でのネットワークのセキュリティ保護](#)』の第 7 章、「[IP フィルタファイアウォールの構成](#)」で説明しているようにループバックフィルタリングをオンにすることによって、非大域ゾーンで Oracle Solaris IP フィルタを有効にできます。

Oracle Solaris IP フィルタは、オープンソースの IP Filter ソフトウェアを基にしています。

共有 IP ゾーン内の IP ネットワークマルチパス

Oracle Solaris では、IP ネットワークマルチパス (IPMP) が複数のインタフェースを同一の IP リンク上でグループ化することによって、継続的なネットワーク可用性を確保します。これらのベースとなるインタフェースは、そのいずれかに障害が発生した場合でもネットワークを引き続き利用できるように互いにバックアップします。IPMP

も、複数のインタフェースを保持するシステムについて、パケットの負荷分散を提供します。

IPMP は、次の方法で Oracle Solaris に実装されます。

- net0、net1、net2 などの複数のインタフェースから、IPMP インタフェース `ipmp0` が構成されます。
- IPMP インタフェース `ipmp0` は、データアドレスと呼ばれる複数の IP アドレスで構成されます。これらのアドレスは、ネットワークトラフィックのホストに使用されます。
- IP アドレスは、ベースとなるインタフェース `netN` 上に直接構成することもできます。これらのアドレスは、ネットワークトラフィックには使用されませんが、ベースとなるインタフェースに障害が発生しているかどうかを判断する障害検出に使用されます。したがって、ベースとなるインタフェース上のこれらの IP アドレスは検査用アドレスと呼ばれます。

データアドレスは `ipmp0` に存在するので、ベースとなる IP インタフェースの 1 つに障害が発生した場合でも、ネットワークを引き続き利用できます。トラフィックは、`ipmp0` の別のアドレスを使用して引き続き送られます。

すべてのネットワーク構成タスクと同様に、IPMP は大域ゾーンで構成します。そのあとで非大域ゾーンに機能を拡張します。機能の拡張は、IPMP インタフェースのデータアドレスの 1 つをゾーンに割り当てることによって行います。

指定された非大域ゾーンで `ipadm` コマンドを使用すると、ゾーンに関連するインタフェースだけが表示されます。

[170 ページの「IP ネットワークマルチパス機能を共有 IP 非大域ゾーンに拡張する方法」](#)を参照してください。ゾーンの構成手順については、[22 ページの「ゾーンの構成方法」](#)を参照してください。IPMP の機能、コンポーネント、および使用法の詳細は、『[Oracle Solaris 11.3 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理](#)』の第 2 章、「[IPMP の管理について](#)」を参照してください。

排他的 IP 非大域ゾーンにおけるネットワーク

排他的 IP ゾーンには、独自の IP 関連の状態があります。ゾーンの構成時に、独自のデータリンクセットがゾーンに割り当てられます。

パケットは、物理リンクに送信されます。その後、Ethernet スイッチや IP ルーターなどのデバイスがパケットを宛先へ転送します。この送信先は、送信元と同じマシン上の別のゾーンである可能性もあります。

仮想リンクの場合、パケットは最初に仮想スイッチに送信されます。宛先のリンクが同じデバイス(同じ物理リンクまたは `etherstub` 上の VNIC など)を経由する場合、

パケットは宛先の VNIC に直接送信されます。それ以外の場合、パケットは VNIC のベースとなる物理リンクに送信されます。

排他的 IP 非大域ゾーンで使用できる機能については、『[Oracle Solaris ゾーン構成リソース](#)』の「[排他的 IP 非大域ゾーン](#)」を参照してください。

排他的 IP ゾーンの区分化

排他的 IP ゾーンはそれぞれ個別の TCP/IP スタックを持っているため、下位のデータリンク層まで分離されます。排他的 IP ゾーンには、1 つ以上のデータリンク名が大域管理者によって割り当てられます。データリンク名は、NIC または NIC 上の VLAN の場合があります。ゾーン管理者は、大域ゾーンの場合と同じ柔軟性とオプションで、これらのデータリンクの IP を構成できます。

排他的 IP データリンクインタフェース

1 つのデータリンク名は、1 つのゾーンだけに割り当てる必要があります。

`dladm show-link` コマンドを使用して、実行中のゾーンに割り当てられているデータリンクを表示できます。

```
sol-t2000-10{pennyc}1: dladm show-link
LINK          CLASS    MTU    STATE    OVER
vsw0          phys    1500   up       --
net0          phys    1500   up       --
netg2         phys    1500   up       --
netg1         phys    1500   up       --
netg3         phys    1500   up       --
zoneA/net0    vnic    1500   up       net0
zoneB/net0    vnic    1500   up       net0
aggr1         aggr    1500   up       net2 net3
vnic0         vnic    1500   up       net1
zoneA/vnic0   vnic    1500   up       net1
vnic1         vnic    1500   up       net1
zoneB/vnic1   vnic    1500   up       net1
vnic3         vnic    1500   up       aggr1
vnic4         vnic    1500   up       aggr1
zoneB/vnic4   vnic    1500   up       aggr1
```

詳細は、[dladm\(1M\)](#) を参照してください。

同一マシン上の排他的 IP ゾーン間の IP トラフィック

排他的 IP ゾーン間での IP パケットの内部ループバックはありません。すべてのパケットが下位のデータリンクまで送信されます。これは通常、パケットがネットワークインタフェース上に送信されることを意味します。その後、Ethernet スイッチや IP

ルーターなどのデバイスがパケットを宛先へ転送します。この送信先は、送信元と同じマシン上の別のゾーンである可能性もあります。

排他的 IP ゾーンにおける Oracle Solaris IP フィルタ

排他的 IP ゾーンでは、大域ゾーンのものと同じ IP フィルタ機能を使用できます。排他的 IP ゾーンでの IP フィルタの構成方法も大域ゾーンと同じです。

排他的 IP ゾーン内の IP ネットワークマルチパス

IP ネットワークマルチパス (IPMP) は、同一の IP リンク上に複数のインタフェースを保持するシステムで、物理インタフェースの障害検出および透過的なネットワークアクセスフェイルオーバーを提供します。IPMP も、複数のインタフェースを保持するシステムについて、パケットの負荷分散を提供します。

データリンク構成は、大域ゾーン内で行われます。最初に、`zonecfg` を使用して、複数のデータリンクインタフェースをゾーンに割り当てます。複数のデータリンクインタフェースを同じ IP サブネットに接続する必要があります。その後、排他的 IP ゾーン内からゾーン管理者が IPMP を構成することができます。

非大域ゾーンでのデバイスの使用

あるゾーンのプロセスが別のゾーンで実行中のプロセスに干渉することがないように、ゾーン内部で利用可能なデバイスセットには制限が課されています。たとえば、ゾーン内のプロセスが、カーネルメモリーおよびルートディスクの内容を変更することはできません。このため、デフォルトでは、ゾーン内で安全に利用可能であると見なされる特定の仮想デバイスだけを使用できます。`zonecfg` ユーティリティーを使用すると、利用可能なデバイスを特定のゾーンに追加できます。

`/dev` および `/devices` 名前空間

`devfs(7FS)` のマニュアルページで説明されている `devfs` ファイルシステムは、Oracle Solaris システムによって `/devices` の管理に使用されます。この名前空間内の各要素は、ハードウェアデバイス、仮想デバイス、またはネクサスデバイスへの物理パスを表します。名前空間には、デバイスツリーが反映されます。したがって、ファイルシステムは、ディレクトリおよびデバイス特殊ファイルの階層により生成されます。

デバイスは相対 `/dev` 階層に従ってグループ化されます。たとえば、大域ゾーンの `/dev` の下のすべてのデバイスは、大域ゾーンのデバイスとしてグループ化されます。

非大域ゾーンの場合、デバイスはゾーンのルートパスの下の `/dev` ディレクトリでグループ化されます。各グループは、`/dev` ディレクトリの下にマウントされた `/dev` ファイルシステムのインスタンスです。したがって、大域ゾーンのデバイスは `/dev` の下にマウントされ、`my-zone` という非大域ゾーンのデバイスは `/my-zone/root/dev` の下にマウントされます。

`/dev` ファイル階層は、`dev` ファイルシステム ([dev\(7FS\)](#)) のマニュアルページに記載によって管理されます。



注意 - `/devices` パス名に依存するサブシステムは、非大域ゾーンで実行できません。`/dev` パス名を使用するようにサブシステムを更新する必要があります。



注意 - 非大域ゾーンに `/dev/zvol` 内のデバイスと一致するデバイスリソースがある場合、非大域ゾーン内で名前空間の競合が発生する可能性があります。詳細は、[dev\(7FS\)](#) のマニュアルページを参照してください。

排他使用のデバイス

デバイスを特定のゾーンに割り当てるが必要な場合があります。特権のないユーザーがブロックデバイスにアクセスできるようにすると、これらのデバイスの使用が許可されて、システムパニックやバスリセットなどの不具合が生じる場合があります。この種の割り当てを行う前に、次の点を考慮してください。

- SCSI テープデバイスを特定のゾーンに割り当てる前に、[sgen\(7D\)](#) のマニュアルページを参照してください。
- 物理デバイスを複数のゾーンに配置する場合、ゾーン間に隠れたチャンネルが作成される場合があります。大域ゾーンアプリケーションでこの種のデバイスを使用すると、非大域ゾーンによるデータ整合性の損失や、データの破壊が発生する危険があります。

デバイスドライバの管理

[modinfo\(1M\)](#) のマニュアルページに記載されているように、非大域ゾーン内で `modinfo` コマンドを使用して、ロードされたカーネルモジュールのリストを検査できます。

カーネル、デバイス、およびプラットフォームの管理に関係した大半の操作は、非大域ゾーンの内部では機能しません。これは、プラットフォームハードウェア構成を変更すると、ゾーンのセキュリティーモデルに違反するためです。これらの操作には、次のことが含まれます。

- ドライバの追加および削除

- カーネルモジュールの明示的な読み込みおよび読み込み解除
- 動的再構成 (DR) 操作の開始
- 物理プラットフォームの状態に影響を与える機能の使用

非大域ゾーンで動作しないか、変更されるユーティリティー

非大域ゾーンで動作しないユーティリティー

次のユーティリティーは、通常は使用できないデバイスに依存しているため、ゾーン内では動作しません。

- `add_drv` ([add_drv\(1M\)](#)) のマニュアルページを参照)
- `disks` ([disks\(1M\)](#)) のマニュアルページを参照)
- `prtconf` ([prtconf\(1M\)](#)) のマニュアルページを参照)
- `prtdiag` ([prtdiag\(1M\)](#)) のマニュアルページを参照)
- `rem_drv` ([rem_drv\(1M\)](#)) のマニュアルページを参照)

SPARC: 非大域ゾーンでの使用に合わせて変更されたユーティリティー

`eeeprom` ユーティリティーをゾーン内で使用して、設定を表示できます。このユーティリティーを使用して、設定を変更することはできません。詳細は、[eeeprom\(1M\)](#) および [openprom\(7D\)](#) のマニュアルページを参照してください。

セキュリティを考慮することにより許可されるユーティリティー

`allowed-raw-io` が有効にされている場合、次のユーティリティーをゾーンで使用できます。セキュリティに関する考慮事項を評価する必要があります。デバイスを追加する場合、制限およびセキュリティの懸念事項については、[136 ページの「非大域ゾーンでのデバイスの使用」](#)、[139 ページの「非大域ゾーンでのアプリケーションの実行」](#)、および [141 ページの「非大域ゾーン内の特権」](#) を参照してください。

- `cdrecord` ([cdrecord\(1\)](#)) のマニュアルページを参照してください)。
- `cdrw` ([cdrw\(1\)](#)) のマニュアルページを参照)。
- `rmformat` ([rmformat\(1\)](#)) のマニュアルページを参照)。

非大域ゾーンでのアプリケーションの実行

通常は、すべてのアプリケーションを非大域ゾーンで実行できます。ただし、次のタイプのアプリケーションは、この環境に適さない場合があります。

- システム全体に影響を与える特権付きの操作を実行するアプリケーション。大域システムクロックを設定したり、物理メモリーをロックダウンする操作が、これに該当します。
- /dev/kmem などの、非大域ゾーン内に存在しない特定のデバイスに依存するいくつかのアプリケーション。
- 共有 IP ゾーンで、/dev/ip 内のデバイスに依存するアプリケーション。

非大域ゾーンで使用されるリソース制御

ゾーン内でのリソース管理機能の使用に関する追加情報については、『[Oracle Solaris 11.3 でのリソースの管理](#)』のこの機能について説明した章を参照してください。

リソース管理の章に記載されたリソース制御および属性はすべて、大域ゾーンおよび非大域ゾーンの /etc/project ファイル、NIS マップ、または LDAP ディレクトリサービスで設定できます。指定されたゾーンの設定は、そのゾーンにのみ影響を及ぼします。異なるゾーン内で自動実行中のプロジェクトは、ゾーンごとに別個の制御セットを保持できます。たとえば、大域ゾーン内のプロジェクト A を `project.cpu-shares=10` に設定し、非大域ゾーン内の Project A を `project.cpu-shares=5` に設定できます。それぞれが該当するゾーン内でのみ機能する、rcapd のインスタンスをシステム上で複数実行できます。

ゾーン内部のプロジェクト、タスク、およびプロセスを制御するため、ゾーン内で使用するリソースの制御および属性は、プールおよびゾーン規模のリソース制御に関する追加要件に従います。

1つの非大域ゾーンにはリソースプールを1つだけ関連付けることができますが、プールは特定のゾーンに排他的に割り当てる必要はありません。1つのプール内のリソースを複数の非大域ゾーンが共有してもかまいません。ただし、十分な特権を付与されたプロセスを使って、大域ゾーン内のプロセスを任意のプールにバインドすることが可能です。リソースコントローラ `pooladm` は、大域ゾーン内だけで動作します。大域ゾーン内には、リソースコントローラが動作するプールが複数存在します。`poolstat` ユーティリティを非大域ゾーンで実行すると、そのゾーンに関連付けられているプールの情報だけが表示されます。非大域ゾーンで引数なしで `pooladm` コマンドを実行すると、そのゾーンに関連付けられているプールの情報だけが表示されます。

ゾーン規模のリソース制御が `project` ファイルで設定されている場合、そのリソース制御は有効にはなりません。ゾーン規模のリソース制御は、`zonecfg` ユーティリティを使って設定されます。

ゾーンがインストールされている システムでの公平配分スケジューラ

このセクションでは、ゾーンで公平配分スケジューラ (FSS) を使用方法について説明します。

大域ゾーンまたは非大域ゾーン内の FSS 配分分割

ゾーンの FSS CPU 配分は、階層的で、大域ゾーンおよび非大域ゾーンの配分は、ゾーン規模のリソース制御 `zone.cpu-shares` を使って大域管理者が設定します。次に、そのゾーン内のプロジェクトごとにリソース制御 `project.cpu-shares` を定義して、ゾーン規模の制御で設定された配分をさらに分割できます。

`zonecfg` コマンドを使用してゾーンに配分を割り当てる方法については、[32 ページの「大域ゾーンの `zone.cpu-shares` を設定する方法」](#)を参照してください。`project.cpu-shares` の詳細は、『[Oracle Solaris 11.3 でのリソースの管理](#)』の「[使用可能なリソース制御](#)」を参照してください。配分を一時的に設定する方法を示す手順例については、[174 ページの「ゾーンがインストールされている Oracle Solaris システムでの公平配分スケジューラの使用」](#)も参照してください。

ゾーン間の配分均衡

`zone.cpu-shares` を使用して、FSS 配分を大域ゾーンと非大域ゾーンに割り当てることができます。FSS がシステムのデフォルトのスケジューラになっている場合、配分が割り当てられていないときは、各ゾーンにはデフォルトで 1 つの配分が付与されます。システムに 1 つの非大域ゾーンが存在し、`zone.cpu-shares` を使ってこのゾーンに 2 つの配分を付与する場合、これにより非大域ゾーンが大域ゾーンとの関連で受ける CPU の比率が定義されます。2 つのゾーン間の CPU 比率は 2:1 です。

ゾーンがインストールされているシステムでの拡張アカウントティング

拡張アカウントティングサブシステムを大域ゾーンで実行した場合、非大域ゾーンを含むシステム全体の情報が収集および報告されます。大域管理者は、ゾーンごとのリソース消費を決定することもできます。

拡張アカウントティングサブシステムを使用すると、プロセスおよびタスクに基づいたアカウントティングに対応した異なるアカウントティング設定およびファイルを、ゾーン単位で指定することが可能になります。exact レコードに、プロセスの場合はゾーン

名 EXD PROC ZONENAME のタグを付け、タスクの場合はゾーン名 EXD TASK ZONENAME のタグを付けることができます。アカウントレコードは、大域ゾーンのアカウンティングファイルおよびゾーン単位のアカウンティングファイルに書き込まれます。EXD TASK HOSTNAME、EXD PROC HOSTNAME、および EXD HOSTNAME レコードには、大域ゾーンのノード名の代わりに、プロセスまたはタスクが実行されたゾーンの `uname -n` 値が含まれます。

非大域ゾーン内の特権

プロセスは、特権のサブセットに制限されています。特権を制限することで、ほかのゾーンに影響を及ぼす可能性がある操作がゾーンで実行されないようにします。特権セットにより、特権が付与されたユーザーがゾーン内で実行可能な機能が制限されます。指定されたゾーン内で利用可能な特権のリストを表示するには、`ppriv` ユーティリティーを使用します。

次の表に、Oracle Solaris の特権すべて、およびゾーン内での各特権のステータスを示します。省略可能な特権は、デフォルト特権セットの一部ではありませんが、`limitpriv` プロパティを使って指定できます。必須の特権が、生成される特権セットに含まれている必要があります。禁止された特権を、生成される特権セットに含めることはできません。

表 1 ゾーン内の特権のステータス

特権	ステータス	注意事項
<code>cpc_cpu</code>	オプション	特定の <code>cpc(3CPC)</code> カウンタへのアクセス
<code>dtrace_proc</code>	オプション	<code>fasttrap</code> および <code>pid</code> プロバイダ。 <code>plockstat(1M)</code>
<code>dtrace_user</code>	オプション	<code>profile</code> および <code>syscall</code> プロバイダ
<code>file_flag_set</code>	オプション	プロセスで <code>immutable</code> 、 <code>nounlink</code> 、または <code>appendonly</code> のファイル属性を設定できるようにします。大域ゾーンで <code>immutable</code> とマークするために使用でき、非待機ゾーンはこのファイルを削除できなくなります。
<code>graphics_access</code>	オプション	<code>ioctl(2)</code> による <code>agpgart_io(7I)</code> へのアクセス
<code>graphics_map</code>	オプション	<code>mmap(2)</code> による <code>agpgart_io(7I)</code> へのアクセス
<code>net_rawaccess</code>	共有 IP ゾーンではオプション。 排他的 IP ゾーンではデフォルト。	raw PF_INET/PF_INET6 パケットアクセス
<code>proc_clock_highres</code>	オプション	高解像度タイマーの使用
<code>proc_priocntl</code>	オプション	スケジューリングの制御。 <code>priocntl(1)</code>

特権	ステータス	注意事項
sys_ipc_config	オプション	IPC メッセージキューのバッファサイズ の引き上げ
dtrace_kernel	禁止	現在、未サポート
proc_zone	禁止	現在、未サポート
sys_config	禁止	現在、未サポート
sys_devices	禁止	現在、未サポート
sys_dl_config	禁止	現在、未サポート
sys_linkdir	禁止	現在、未サポート
sys_net_config	禁止	現在、未サポート
sys_res_config	禁止	現在、未サポート
sys_smb	禁止	現在、未サポート
sys_suser_compat	禁止	現在、未サポート
file_read	必須、デフォルト	プロセスが、プロセスの読み取り権を許可 するアクセス権または ACL を持つファイル またはディレクトリを読み取れるようにし ます
file_write	必須、デフォルト	プロセスが、プロセスの書き込み権を許可 するアクセス権または ACL を持つファイル またはディレクトリを書き込めるようにし ます
net_access	必須、デフォルト	プロセスが TCP、UDP、SDP、または SCTP ネットワークエンドポイントを開けるよう にします
proc_exec	必須、デフォルト	init(1M) の起動に使用
proc_fork	必須、デフォルト	init(1M) の起動に使用
sys_mount	必須、デフォルト	必須ファイルシステムのマウントに必要
sys_flow_config	必須、排他的 IP ゾーンでは デフォルト 共有 IP ゾーンでは禁止	フローの構成に必要
sys_ip_config	必須、排他的 IP ゾーンでは デフォルト 共有 IP ゾーンでは禁止	ゾーンのブートおよび排他的 IP ゾーンの IP ネットワークの初期化に必要
sys_iptun_config	必須、排他的 IP ゾーンでは デフォルト 共有 IP ゾーンでは禁止	IP トンネルリンクの構成
contract_event	デフォルト	契約ファイルシステムで使用
contract_identity	デフォルト	プロセス規約テンプレートのサービス FMRI 値の設定
contract_observer	デフォルト	UID とは無関係な契約観察
file_chown	デフォルト	ファイル所有権の変更
file_chown_self	デフォルト	所有するファイルの所有者/グループの変更

特権	ステータス	注意事項
file_dac_execute	デフォルト	モード/ACL に依存しない実行アクセス
file_dac_read	デフォルト	モード/ACL に依存しない読み取りアクセス
file_dac_search	デフォルト	モード/ACL に依存しない検索アクセス
file_dac_write	デフォルト	モード/ACL に依存しない書き込みアクセス
file_link_any	デフォルト	所有者に依存しないリンクアクセス
file_owner	デフォルト	所有者に依存しないその他のアクセス
file_setid	デフォルト	setid、setgid、setuid ファイルのアクセス権の変更
ipc_dac_read	デフォルト	モードに依存しない IPC 読み取りアクセス
ipc_dac_write	デフォルト	プロセスが、通常であればアクセス権ビットがプロセスの書き込み権を許可しない System V IPC メッセージキュー、セマフォセット、または共有メモリーセグメントを書き込めるようにします
ipc_dac_owner	デフォルト	モードに依存しない IPC 書き込みアクセス
ipc_owner	デフォルト	モードに依存しないその他の IPC アクセス
net_icmpaccess	デフォルト	ICMP パケットアクセス: ping(1M)
net_observability	デフォルト	プロセスが、ネットワークトラフィックの受信用のデバイスを開けるようにします。トラフィックの送信は禁止されています
net_privaddr	デフォルト	特権ポートへのバインド
proc_audit	デフォルト	監査レコードの生成
proc_chroot	デフォルト	root ディレクトリの変更
proc_info	デフォルト	プロセスの検査
proc_lock_memory	デフォルト	メモリーのロック。shmctl(2) および mlock(3C) この特権がシステム管理者によって非大域ゾーンに割り当てられている場合、ゾーンがすべてのメモリーをロックするのを防ぐために zone.max-locked-memory リソース制御の設定も検討してください。
proc_owner	デフォルト	所有者に依存しないプロセス制御
proc_session	デフォルト	セッションに依存しないプロセス制御
proc_setid	デフォルト	ユーザー/グループ ID の任意設定
proc_taskid	デフォルト	呼び出し元へのタスク ID の割り当て
sys_acct	デフォルト	アカウントの管理
sys_admin	デフォルト	単純なシステム管理タスク
sys_audit	デフォルト	監査の管理
sys_nfs	デフォルト	NFS クライアントのサポート
sys_ppp_config	排他的 IP ゾーンではデフォルト	PPP (sppp) インタフェースの作成および削除、PPP トンネル (sppptun) の構成

特権	ステータス	注意事項
	共有 IP ゾーンでは禁止	
sys_resource	デフォルト	リソース制限の操作
sys_share	デフォルト	ファイルシステムの共有に必要な sharefs システムコールを許可します。ゾーン内の NFS 共有を防止するために、ゾーン内で特権を禁止にすることができます。
sys_time	デフォルト	システム時間の操作。xntp(1M)

次の表に、Oracle Solaris Trusted Extensions の特権すべて、および各特権のゾーン内のステータスを示します。省略可能な特権は、デフォルト特権セットの一部ではありませんが、limitpriv プロパティを使って指定できます。

注記 - Oracle Trusted Solaris 特権が解釈されるのは、システムが Oracle Trusted Extensions を使って構成されている場合だけです。

表 2 ゾーン内での Oracle Solaris Trusted Extensions の特権のステータス

Oracle Solaris Trusted Extensions の特権	ステータス	注意事項
file_downgrade_sl	オプション	ファイルまたはディレクトリの機密ラベルを、既存の機密ラベルを優先する機密ラベルに設定します。
file_upgrade_sl	オプション	ファイルまたはディレクトリの機密ラベルを、既存の機密ラベルよりも優先される機密ラベルに設定します。
sys_trans_label	オプション	機密ラベルの制御下でないラベルの変換
win_colormap	オプション	カラーマップ制限のオーバーライド
win_config	オプション	X サーバーにより常時保持されるリソースの構成または破棄
win_dac_read	オプション	クライアントのユーザー ID が所有していないウィンドウリソースからの読み取り
win_dac_write	オプション	クライアントのユーザー ID が所有していないウィンドウリソースへの書き込みまたは作成
win_devices	オプション	入力デバイスでの操作の実行
win_dga	オプション	ダイレクトグラフィックスアクセス X プロトコル拡張機能の使用。フレームバッファ特権が必要
win_downgrade_sl	オプション	ウィンドウリソースの機密ラベルを、既存ラベルの制御下にある新規ラベルに変更
win_fontpath	オプション	フォントパスの追加
win_mac_read	オプション	クライアントのラベルを制御するラベルを使用した、ウィンドウリソースからの読み取り

Oracle Solaris Trusted Extensions の特権	ステータス	注意事項
win_mac_write	オプション	クライアントのラベルと同等ではないラベルを使用した、ウィンドウリソースへの書き込み
win_selection	オプション	確認者の介入なしでの要求データの移動
win_upgrade_sl	オプション	ウィンドウリソースの機密ラベルを、既存ラベルの制御下でない新規ラベルに変更
net_bindmlp	デフォルト	マルチレベルポート (MLP) へのバインドの許可
net_mac_aware	デフォルト	NFS を使用した読み取りの許可

非大域ゾーン構成内の特権を変更する方法については、[22 ページの「ゾーンを構成、検証、および確定する」](#)を参照してください。

特権セットを検査する方法については、[157 ページの「ppriv ユーティリティーの使用」](#)を参照してください。特権の詳細は、[ppriv\(1\)](#) のマニュアルページおよび『Solaris のシステム管理 (セキュリティサービス)』を参照してください。

ゾーン内での IP セキュリティアーキテクチャーの使用

IP データグラムの保護を提供する IPsec (Internet Protocol Security Architecture) については、『Oracle Solaris 11.3 でのネットワークのセキュリティ保護』の「IPsec リファレンス」に記載されています。IKE (Internet Key Exchange) プロトコルを使用して、認証および暗号化に必要な鍵材料が自動的に管理されます。

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

共有 IP ゾーン内の IP セキュリティアーキテクチャー

IPsec は、大域ゾーン内で使用できます。ただし、非大域ゾーン内では、IPsec で IKE を使用することはできません。そのため、大域ゾーンで Internet Key Exchange (IKE) プロトコルを使用して、非大域ゾーンの IPsec キーおよび IPsec ポリシーを管理する必要があります。構成中の非大域ゾーンに対応するソースアドレスを使用します。

排他的 IP ゾーン内の IP セキュリティアーキテクチャー

IPsec は、排他的 IP ゾーン内で使用できます。

ゾーン内での Oracle Solaris 監査の使用

監査記録には、システムへのログインやファイルへの書き込みなどのイベントが記載されます。Oracle Solaris 監査は、ゾーンを実行しているシステムで次の 2 つの監査モデルを提供します。

- すべてのゾーンを大域ゾーンから一元的に監査します。このモデルは、すべてのゾーンが大域ゾーンによって管理される場合 (たとえば、ゾーンを使用してサービスの隔離を実現する場合) に使用されます。
- 各ゾーンを大域ゾーンから独立して監査します。このモデルは、各ゾーンが別々に管理される場合 (たとえば、ゾーンごとにサーバーの統合を実現する場合) に使用されます。

Oracle Solaris 監査は、『[Oracle Solaris 11.3 での監査の管理](#)』の第 1 章、「[Oracle Solaris での監査について](#)」に記載されています。監査に関連したゾーンの考慮事項については、『[Oracle Solaris 11.3 での監査の管理](#)』の「[Oracle Solaris Zones を使用したシステムの監査](#)」および『[Oracle Solaris 11.3 での監査の管理](#)』の「[ゾーンでの監査サービスの構成](#)」を参照してください。詳細は、[auditconfig\(1M\)](#)、[auditreduce\(1M\)](#)、[usermod\(1M\)](#)、および [user_attr\(4\)](#) のマニュアルページも参照してください。

注記 - 一時的に有効にされるが、リポジトリには設定されない監査ポリシーを使用することもできます。

詳細は、『[Oracle Solaris 11.3 での監査の管理](#)』の「[監査ポリシーを変更する方法](#)」に続いて記載されている例を参照してください。

ゾーン内のコアファイル

`coreadm` コマンドを使用して、異常終了するプロセスにより生成されるコアファイルの名前と場所を指定できます。`%z` 変数を指定することで、プロセスが実行されたゾーンの `zonename` を含むコアファイルパスを生成できます。パス名は、ゾーンのルートディレクトリに対する相対パスです。

詳細は、[coreadm\(1M\)](#) および [core\(4\)](#) のマニュアルページを参照してください。

非大域ゾーン内での DTrace の実行

`dtrace_proc` および `dtrace_user` 特権のみを必要とする DTrace プログラムは、非大域ゾーンで実行できます。非大域ゾーンで使用できる特権のセットにこれらの特権を追加するには、`zonecfg limitpriv` プロパティを使用します。手順については、164 ページの「DTrace を使用する方法」を参照してください。

`dtrace_proc` によってサポートされるプロバイダは、`fasttrap` と `pid` です。`dtrace_user` によってサポートされるプロバイダは、`profile` と `syscall` です。DTrace のプロバイダおよびアクションの有効範囲は、ゾーンに制限されます。

詳細は、141 ページの「非大域ゾーン内の特権」も参照してください。

非大域 Oracle Solaris ゾーンでの時間値の設定

非大域ゾーンでは、大域ゾーン内の値とは異なる時間値を設定できます。ゾーンで時間を設定する機能は、デフォルトの `sys_time` 権限を通じて利用できます。この特権により、非大域ゾーンのプロセスは、そのゾーンの `global-time` プロパティの値 (設定されている場合) に応じて、仮想ゾーン時間またはシステム時間のどちらかを設定できます。非大域ゾーンで異なる時間を設定する機能は、大域ゾーンで行われた時間の変更から独立しているわけではありません。大域ゾーンで時間が変更された場合は、非大域ゾーンの時間が同じ量で相殺されます。

NTP (Network Time Protocol) はどのゾーンからでも実行でき、それにより、そのコマンドが実行されているゾーンだけが影響を受けます。時間が異なる非大域ゾーンを含むシステム全体で NTP を実行している場合は、大域ゾーンで NTP を実行してください。大域ゾーンで NTP を実行すると、オフセットでのみ動作している非大域ゾーンのすべてのクロックが同期されます。大域ゾーンで NTP によるクロック調整を通じた時間の変更を行うと、その効果は非大域ゾーンにも及びます。

`global-time` が `false` に設定されているゾーンの側で NTP を実行した場合は、`ntp_adjtime` および `adjtime` システムコールを使ってゾーンの時間を修正できません。`global-time` が `false` に設定されている場合、NTP は時間の設定を通じてクロックを指定された値に調整することでタイミングを合わせ、第 2 レベルの精度で時間の同期を維持します。NTP は、時間の同期を維持するためにゾーンの時間を前後に調整できます。

`global-time` が `true` に設定されているゾーンの側で NTP を実行した場合は、`ntp_adjtime` および `adjtime` システムコールを使ってシステムクロックを調整できます。その後、ゾーンの側で NTP を実行してシステム時間の正確な同期を維持する機能が保持されます。

`global-time` プロパティの詳細は、『Oracle Solaris ゾーン構成リソース』を参照してください。時間の設定方法の詳細は、[date\(1\)](#) を参照してください。権限の詳細

は、[141 ページの「非大域ゾーン内の特権」](#)を参照してください。adjtime および ntp_adjtime の詳細は、[adjtime\(2\)](#) および [ntp_adjtime\(2\)](#) のマニュアルページを参照してください。

ゾーンがインストールされている Oracle Solaris システムのバックアップについて

非大域ゾーンを個別にバックアップしたり、大域ゾーンからシステム全体をバックアップしたりできます。

ループバックファイルシステムのディレクトリのバックアップ

非大域ゾーン内からループバックファイルシステム (lofs) をバックアップしないでください。

非大域ゾーン内から読み取り/書き込みループバックファイルシステムをバックアップおよび復元する場合、これらのファイルシステムは大域ゾーンおよび読み取り/書き込みマウントされているほかのすべてのゾーンからも書き込むことができます。複数のコピーを防止するために、これらのファイルシステムは大域ゾーンからのみバックアップおよび復元します。

大域ゾーンからのシステムのバックアップ

次のような場合は、大域ゾーンからバックアップを実行することをお勧めします。

- 非大域ゾーンの構成をアプリケーションデータと一緒にバックアップする場合。
- 障害から回復することが最も重要である場合。使用しているゾーンのルートファイルシステムおよびそれらの構成データ、使用している大域ゾーン内のデータなど、システム上のすべてまたはほぼすべての情報を復元する必要がある場合。このような場合は、大域ゾーンでバックアップを実行してください。
- 市販のネットワークバックアップソフトウェアを使用する場合。

注記 - ネットワークバックアップソフトウェアを使用するときには、可能であれば、継承された lofs ファイルシステムはすべてスキップするように構成することをお勧めします。バックアップは、ゾーンとそのアプリケーションがバックアップ対象のデータを休止させた状態のときに、行うことをお勧めします。

システム上の非大域ゾーンを個別にバックアップ

次のような場合は、非大域ゾーン内でバックアップを実行することをお勧めします。

- 非大域ゾーンの管理者が、重大度の低い障害から回復する機能、またはゾーンに固有のアプリケーションデータまたはユーザーデータを復元する機能を必要とする場合。
- ゾーン内で動作する特定のアプリケーションまたはサービスのバックアップソフトウェアを使用する場合。ディレクトリパスやインストール済みソフトウェアなどのアプリケーション環境が大域ゾーンと非大域ゾーンとの間で異なっている場合には、バックアップソフトウェアを大域ゾーンから実行するのが困難な場合があります。

アプリケーションが非大域ゾーンごとのバックアップスケジュールに基づいてスナップショットを実行し、そのバックアップデータを大域ゾーンからエクスポートした書き込み可能なディレクトリに格納できる場合には、大域ゾーンの管理者は大域ゾーンからバックアップする処理の中でそれら個々のバックアップデータを個別に選択できます。

Oracle Solaris 統合アーカイブのバックアップの作成

zonecfg および zoneadm コマンドを使用すると、統合アーカイブファイルから直接、新しいゾーンを構成およびインストールできます。統合アーカイブファイルには、ゾーン構成とゾーンデータの両方が含まれています。移行先サーバー上で、アーカイブからゾーンを構成およびインストールできます。

システムおよびゾーンのクローニングや回復のための使用方法を含む統合アーカイブの詳細は、『[Oracle Solaris 11.3 でのシステム復旧とクローン](#)』を参照してください。

非大域ゾーン内でバックアップするデータの決定

非大域ゾーン内のデータは、すべてバックアップできます。ゾーンの構成が頻繁に変更されない場合には、アプリケーションデータだけをバックアップすることもできます。

アプリケーションデータのためのバックアップ

アプリケーションデータがファイルシステムの特定の場所に格納されている場合には、このデータだけを定期的にバックアップすることもできます。ゾーンのルートファイルシステムはそれほど頻繁には変更されないため、頻繁にバックアップする必要がない場合もあります。

アプリケーションファイルがどこに格納されているかを確認する必要があります。アプリケーションファイルは次のような場所に格納されている可能性があります。

- ユーザーのホームディレクトリ
- /etc (構成データファイルの場合)
- /var

アプリケーション管理者がデータの格納場所を認識している場合には、ゾーンごとに書き込み可能ディレクトリを割り当てるように、システムを作成することもできます。バックアップがゾーンごとに格納されるので、大域管理者または適切な承認を付与されたユーザーは、その場所をシステム上のバックアップ対象の1つとして選択することができます。

一般的なデータベースバックアップ操作

データベースアプリケーションデータがデータベース固有のディレクトリに存在していない場合には、次の規則が適用されます。

- 最初にデータベースが安定した状態にあることを確認します。
データベースが休止している必要があります。内部バッファのデータがディスクにフラッシュされるためです。大域ゾーンからバックアップを開始する前に、非大域ゾーン内のデータベースが安定した状態になっていることを確認してください。
- ゾーンごとに、ファイルシステム機能を使用してデータのスナップショットを作成してから、そのスナップショットを大域ゾーンから直接バックアップします。
このようにすると、バックアップのための時間が短縮されるだけでなく、すべてのゾーンのクライアントやモジュールをバックアップする必要がなくなります。

テープによるバックアップ

非大域ゾーンだけが使用するファイルシステムについては、そのゾーンにとって都合のよい時間帯に、アプリケーションがわずかに休止している時間を利用して、スナップショットを作成することもできます。それらのスナップショットは、アプリケーションがサービスに戻ったあとに大域ゾーンからバックアップしてテープに格納できます。

この方法により、次の利点が得られます。

- 必要なテープデバイスが少なくてすみます。
- 非大域ゾーンの間で調整する必要がなくなります。
- デバイスを直接ゾーンに割り当てる必要がないため、セキュリティ機能が向上します。

- 大域ゾーンでシステム管理を続けることができるので、通常はこの方法をお勧めします。

非大域ゾーンの復元について

大域管理者または適切な承認を付与されたユーザーは、大域ゾーンから実行したバックアップを復元するときには、関係するゾーンを再インストールしてから、そのゾーンのファイルを復元できます。この処理を行うときには、ゾーンが次の状態である必要があります。

- 復元するゾーンの構成が、バックアップしたときと同じ構成であること。
- バックアップしてからゾーンを復元するまでの間に、大域ゾーンをアップデートしていないこと。

これらの前提を満たしていない場合は、一部のファイルが復元によって上書きされ、手作業でマージしなければならないことがあります。

注記 - 大域ゾーンのすべてのファイルシステムが失われた場合には、大域ゾーンのすべてのファイルを復元すると、非大域ゾーンも復元されます。ただし、非大域ゾーンの各ルートファイルシステムがバックアップに含まれている必要があります。

ゾーンがインストールされているシステムで使用するコマンド

表3で示すコマンドは、ゾーン機能に対する主な管理インタフェースを示しています。

表 3 ゾーン管理およびモニターに使用するコマンド

コマンドリファレンス	説明
zlogin(1)	非大域ゾーンにログインします
zonename(1)	現在のゾーンの名前を出力します
zonestat(1)	ゾーンのリソース使用率の監視に使用します
zoneadm(1M)	システムのゾーンを管理します
zonecfg(1M)	ゾーン構成の設定に使用されます
getzoneid(3C)	ゾーン ID と名前とのマッピングに使用されます
zones(5)	ゾーン機能の説明を提供します
zcons(7D)	ゾーンコンソールのデバイスドライバ

zoneadm デーモンは、ゾーンの仮想プラットフォームを管理する基本プロセスです。zoneadm デーモンのマニュアルページは、zoneadm(1M) です。このデーモンは、プログラミングインタフェースの構成要素ではありません。

次の表に示すコマンドは、リソース上限デーモンとともに使用されます。

表 4 rcapd とともに使用されるコマンド

コマンドリファレンス	説明
rcapstat(1)	上限が定義されたプロジェクトのリソース使用効率をモニターします。
rcapadm(1M)	リソース上限デーモンを構成します。構成済みのリソース上限デーモンの現在のステータスを表示します。リソース上限制御を有効または無効にします。
rcapd(1M)	リソース上限デーモン。

次の表で示すコマンドは、ゾーンがインストールされている Oracle Solaris システムでできるように変更されています。これらのコマンドには、ゾーンに固有のオプションが用意されています。指定するオプションによって異なる情報が表示されず。コマンドは、マニュアルページのセクション別に記載されています。

表 5 ゾーンがインストールされている Oracle Solaris システムで使用するために変更されたコマンド

コマンドリファレンス	説明
ipcrm(1)	-z zone オプションが追加されました。このオプションは、大域ゾーン内でコマンドを実行する場合にのみ有用です。
ipcs(1)	-z zone オプションが追加されました。このオプションは、大域ゾーン内でコマンドを実行する場合にのみ有用です。
pgrep(1)	-z zoneidlist オプションが追加されました。このオプションは、大域ゾーン内でコマンドを実行する場合にのみ有用です。
ppriv(1)	-l オプションとともに使用する式 zone が追加されました。これを使用すると、現在のゾーン内で使用可能なすべての特権が一覧表示されます。また、zone に -v オプションを指定して、冗長出力を取得できます。
prioctl(1)	idlist と -i idtype でゾーン ID を使用することで、プロセスを指定できます。prioctl -i zoneid コマンドを使用すると、実行中のプロセスを非大域ゾーン内の別のスケジューリングクラスに移動できます。
proc(1)	ptree だけに -z zone オプションが追加されました。このオプションは、大域ゾーン内でコマンドを実行する場合にのみ有用です。
ps(1)	-o オプションが使用する、認識される format 名リストに、zonename と zoneid が追加されました。 指定したゾーン内のプロセスだけを一覧表示するため、-z zonelist が追加されました。ゾーンの指定には、ゾーン名またはゾーン ID を使用できません。このオプションは、大域ゾーン内でコマンドを実行する場合にのみ有用です。 プロセスに関連するゾーンの名前を出力するため、-z が追加されました。名前は、追加された列ヘッダー ZONE の下に出力されます。
renice(1)	有効な引数をリスト表示するため、-i オプションとともに使用する zoneid が追加されました。

コマンドリファレンス	説明
sar(1)	プール機能が有効な非大域ゾーン内で実行する際、 <code>-b</code> 、 <code>-c</code> 、 <code>-g</code> 、 <code>-m</code> 、 <code>-p</code> 、 <code>-u</code> 、 <code>-w</code> 、および <code>-y</code> オプションを指定すると、ゾーンのバインド先プールのプロセッサセット内のプロセッサの値だけが表示されます。
auditconfig(1M)	<code>zonename</code> トークンが追加されました。
auditreduce(1M)	<code>-z zone-name</code> オプションが追加されました。ゾーンの監査ログを取得する機能が追加されました。
coreadm(1M)	プロセスが実行されたゾーンを識別するための変数 <code>%z</code> が追加されました。
df(1M)	すべての可視ゾーン内のマウントを表示する <code>-z</code> オプションが追加されました。このオプションは、非大域ゾーンでは効果がありません。
dladm(1M)	デフォルトのコマンド出力にゾーン列を追加する <code>-z</code> オプションが <code>show</code> サブコマンドに追加されました。ゾーン列は、現在、リソースが割り当てられているゾーンを示します。
dlstat(1M)	デフォルトのコマンド出力にゾーン列を追加する <code>-z</code> オプションが <code>show</code> サブコマンドに追加されました。ゾーン列は、現在、リソースが割り当てられているゾーンを示します。
fsstat(1M)	<p>ゾーン別のファイルシステムアクティビティーについて報告する <code>-z</code> オプションが追加されました。複数の <code>-z</code> オプションを使用して、選択したゾーンのアクティビティーをモニターできます。このオプションは、<code>fstypes</code> ではなく <code>mountpoints</code> のみをモニターする場合に使用しても効果はありません。</p> <p>指定した <code>fstypes</code> のファイルシステムアクティビティーのすべてのゾーンにわたる集計を報告する <code>-A</code> オプションが追加されました。これは、<code>-z</code> または <code>-z</code> オプションがどちらも使用されなかった場合のデフォルトの動作です。<code>-A</code> オプションは、<code>fstypes</code> ではなく <code>mountpoints</code> のみをモニターする場合に使用しても効果はありません。</p> <p><code>-A</code> オプションを <code>-z</code> または <code>-z</code> オプションとともに使用すると、指定した <code>fstypes</code> のすべてのゾーンにわたる集計が別個の行に表示されます。</p> <p>システム上のすべてのゾーンのファイルシステムアクティビティーを報告する <code>-Z</code> オプションが追加されました。このオプションは、<code>-z</code> オプションとともに使用しても効果はありません。このオプションは、<code>fstypes</code> ではなく <code>mountpoints</code> のみをモニターする場合に使用しても効果はありません。</p>
iostat(1M)	プール機能が有効な非大域ゾーン内で実行すると、ゾーンのバインド先プールのプロセッサセット内のプロセッサ情報だけが提供されます。
ipadm(1M)	インターネットプロトコルネットワークインタフェースおよび TCP/IP 調整可能パラメータの構成 <code>from-gz</code> タイプは、非大域ゾーンでのみ表示され、非大域排他的 IP ゾーンに構成されている <code>allowed-address</code> プロパティーに基づいて大域ゾーンから構成されたアドレスであることを示します。 <code>zone</code> アドレスプロパティーは、 <code>allowed-address</code> によって参照されるすべてのアドレスが配置されるゾーンを指定します。ゾーンは、共有 IP ゾーンとして構成する必要があります。
kstat(1M)	大域ゾーンで実行すると、すべてのゾーンの <code>kstat</code> が表示されます。非大域ゾーンで実行すると、一致する <code>zoneid</code> を持つ <code>kstat</code> だけが表示されます。

ゾーンがインストールされているシステムで使用するコマンド

コマンドリファレンス	説明
mpstat(1M)	プール機能が有効な非大域ゾーン内で実行すると、ゾーンのバインド先プールのプロセッサセット内のプロセッサの行だけが表示されます。
nnd(1M)	大域ゾーン内で使用された場合、すべてのゾーンの情報を表示します。排他的 IP ゾーンの TCP/IP モジュールに対して <code>nnd</code> を実行すると、そのゾーンの情報だけが表示されます。
netstat(1M)	現在のゾーンのみを表示します。
nfsstat(1M)	現在のゾーンのみを統計情報を表示します。
poolbind(1M)	<code>zoneid</code> リストが追加されました。リソースプールとともにゾーンを使用する方法については、『Oracle Solaris 11.3 でのリソースの管理』の「ゾーンで使用されるリソースプール」も参照してください。
prstat(1M)	<code>-z zoneidlist</code> オプションが追加されました。また、 <code>-Z</code> オプションも追加されました。 プール機能が有効な非大域ゾーンで実行した場合、ゾーンのバインド先プールのプロセッサセット内のプロセッサだけを対象にして、プロセスが使用した最新の CPU 時間の比率が表示されます。 <code>-a</code> 、 <code>-t</code> 、 <code>-T</code> 、 <code>-J</code> 、および <code>-z</code> の各オプションを指定すると、出力にはサイズ列の代わりにスワップ列が表示されます。報告されるスワップは、ゾーンのプロセスと <code>tmpfs</code> マウントで消費されるスワップの合計量です。この値により、各ゾーンで予約されているスワップをモニタリングしやすくなり、適切な <code>zone.max-swap</code> 設定を選択できます。
psrinfo(1M)	非大域ゾーン内で実行した場合、ゾーンで表示可能なプロセッサの情報だけが表示されます。
traceroute(1M)	使用方法が変更されました。非大域ゾーン内から指定した場合、 <code>-F</code> オプションをしても効果はありません。理由は、「断片化しない」というビットが常に設定されているためです。
vmstat(1M)	プール機能が有効な非大域ゾーン内で実行すると、ゾーンのバインド先プールのプロセッサセット内のプロセッサ統計情報だけが報告されます。 <code>-p</code> オプション指定時の出力と、 <code>page</code> 、 <code>faults</code> 、および <code>cpu</code> 報告フィールドに適用されます。
prioctl(2)	<code>P_ZONEID id</code> 引数が追加されました。
processor_info(2)	呼び出し側が非大域ゾーン内にあり、プール機能が有効であるが、プロセッサがゾーンのバインド先プールのプロセッサセット内でない場合、エラーが返されます。
p_online(2)	呼び出し側が非大域ゾーン内にあり、プール機能が有効であるが、プロセッサがゾーンのバインド先プールのプロセッサセット内でない場合、エラーが返されます。
pset_bind(2)	<code>idtype</code> として <code>P_ZONEID</code> が追加されました。 <code>P_MYID</code> 仕様の選択肢にゾーンが追加されました。 <code>EINVAL</code> エラー説明内の有効な <code>idtype</code> リストに <code>P_ZONEID</code> が追加されました。
pset_info(2)	呼び出し側が非大域ゾーン内にあり、プール機能が有効であるが、プロセッサがゾーンのバインド先プールのプロセッサセット内でない場合、エラーが返されます。
pset_list(2)	呼び出し側が非大域ゾーン内にあり、プール機能が有効であるが、プロセッサがゾーンのバインド先プールのプロセッサセット内でない場合、エラーが返されます。

コマンドリファレンス	説明
pset_setattr(2)	呼び出し側が非大域ゾーン内にあり、プール機能が有効であるが、プロセッサがゾーンのバインド先プールのプロセッサセット内でない場合、エラーが返されます。
sysinfo(2)	PRIV_SYS_CONFIG が PRIV_SYS_ADMIN に変更されました。
umount(2)	<i>file</i> が参照しているファイルが絶対パスでない場合、ENOENT が返されます。
getloadavg(3C)	呼び出し側が非大域ゾーン内にあり、プール機能が有効な場合、PS_MYID の psetid を使用して呼び出した場合と動作は同じになります。
getpriority(3C)	ゾーン ID が、指定可能なターゲットプロセスに追加されました。ゾーン ID が EINVAL のエラー説明に追加されました。
priv_str_to_set(3C)	呼び出し側のゾーン内部で使用可能なすべての特権セットで、「zone」文字列が追加されました。
pset_getloadavg(3C)	呼び出し側が非大域ゾーン内にあり、プール機能が有効であるが、プロセッサがゾーンのバインド先プールのプロセッサセット内でない場合、エラーが返されます。
sysconf(3C)	呼び出し側が非大域ゾーン内にあり、プール機能が有効な場合、sysconf(_SC_NPROCESSORS_CONF) および sysconf(_SC_NPROCESSORS_ONLN) は、ゾーンのバインド先プールのプロセッサセット内の合計プロセッサ数およびオンラインのプロセッサ数を返します。
ucred_get(3C)	ucred_getzoneid() 関数が追加されました。この関数は、プロセスのゾーン ID を返します。ただし、ゾーン ID を取得できなかった場合は -1 を返します。
core(4)	n_type: NT_ZONENAME が追加されました。このエントリには、プロセスが実行されていたゾーンの名前を示す文字列が含まれます。
pkginfo(4)	ゾーンのサポート内でオプションパラメータおよび環境変数が提供されるようになりました。
proc(4)	ゾーン内で実行中のプロセスに関する情報を取得する機能が追加されました。
audit_syslog(5)	in<zone name> フィールドが追加されました。このフィールドは、zonename 監査ポリシーが設定されている場合に使用されます。
privileges(5)	プロセスによる追跡またはほかのゾーン内のプロセスへのシグナル送信を可能にする PRIV_PROC_ZONE が追加されました。zones(5) を参照してください。
if_tcp(7P)	ゾーンの ioctl() 呼び出しが追加されました。
cmn_err(9F)	ゾーンパラメータが追加されました。
ddi_cred(9F)	cr の指し示すユーザー資格からゾーン ID を返す、crgetzoneid() が追加されました。

Oracle Solaris ゾーンの管理

この章では、一般的な管理タスクおよびその使用例を示します。

- 157 ページの「`ppriv` ユーティリティーの使用」
- 159 ページの「非大域ゾーンでの `zonestat` ユーティリティーの使用」
- 163 ページの「非大域ゾーン内での `DTrace` の使用」
- 165 ページの「稼働中の非大域ゾーン内でファイルシステムをマウントする」
- 168 ページの「大域ゾーン内の特定のファイルシステムへのアクセス権を非大域ゾーンに追加する」
- 170 ページの「ゾーンがインストールされている Oracle Solaris システムでの IP ネットワークマルチパスの使用」
- 171 ページの「排他的 IP 非大域ゾーンでのデータリンクの管理」
- 174 ページの「ゾーンがインストールされている Oracle Solaris システムでの公平配分スケジューラの使用」
- 176 ページの「ゾーン管理での権利プロファイルの使用」
- 176 ページの「ゾーンがインストールされている Oracle Solaris システムのバックアップ」
- 177 ページの「非大域ゾーンの再作成」

一般的なゾーン管理のトピックについては、第9章「Oracle Solaris ゾーンの管理について」を参照してください。

`ppriv` ユーティリティーの使用

`ppriv` ユーティリティーを使用してゾーンの特権を表示します。

▼ 大域ゾーンでの Oracle Solaris の特権を一覧表示する方法

`ppriv` ユーティリティーを `-l` オプションとともに使用して、システムで使用可能な特権を一覧表示します。

- プロンプトで `ppriv -l zone` と入力し、ゾーンで使用可能な特権セットを表示します。

```
global# ppriv -l zone
```

次のような内容が表示されます。

```
contract_event
contract_observer
cpc_cpu
.
.
```

▼ 非大域ゾーンの特権セットの表示方法

`ppriv` ユーティリティーを `-l` オプションおよび式 `zone` とともに使用して、ゾーンの特権を表示します。

1. 非大域ゾーンにログインします。この例では、`my-zone` という名前のゾーンを使用します。
2. プロンプトで `ppriv -l zone` と入力し、ゾーンで使用可能な特権セットを表示します。

```
my-zone# ppriv -l zone
```

次のような内容が表示されます。

```
contract_event
contract_identity
contract_observer
file_chown
.
.
```

▼ 非大域ゾーンの特権セットを冗長出力で表示する方法

`ppriv` ユーティリティーを `-l` オプション、式 `zone`、および `-v` オプションとともに使用して、ゾーンの特権を一覧表示します。

1. 非大域ゾーンにログインします。この例では、*my-zone* という名前のゾーンを使用します。
2. プロンプトで `ppriv -l -v zone` と入力して、ゾーン内で使用可能な特権セットおよび各特権の説明を出力します。

```
my-zone# ppriv -lv zone
```

次のような内容が表示されます。

```
contract_event
    Allows a process to request critical events without limitation.
    Allows a process to request reliable delivery of all events on
    any event queue.
contract_identity
    Allows a process to set the service FMRI value of a process
    contract template.
contract_observer
    Allows a process to observe contract events generated by
    contracts created and owned by users other than the process's
    effective user ID.
    Allows a process to open contract event endpoints belonging to
    contracts created and owned by users other than the process's
    effective user ID.
file_chown
    Allows a process to change a file's owner user ID.
    Allows a process to change a file's group ID to one other than
    the process' effective group ID or one of the process'
    supplemental group IDs.
.
.
.
```

非大域ゾーンでの zonestat ユーティリティの使用

zonestat ユーティリティは、現在実行中のゾーンの CPU、メモリー、ネットワーク、およびリソース制御の使用効率について報告します。使用例はこのあとにあります。

詳細は、[zonestat\(1\)](#) を参照してください。

zonestat ネットワークコンポーネントは、PHYS、AGGR、Etherstub、および SIMNET データリンクの仮想ネットワーク (VNIC) リソースの使用率をゾーンごとに表示します。その他のデータリンク (ブリッジ、トンネルなど) の情報は、ネットワークユーティリティ ([dladm\(1M\)](#) および [dlstat\(1M\)](#)) のマニュアルページに記載) を使用して取得できます。

ゾーンの統計情報を表示するために、非大域ゾーン内ですべての zonestat オプションおよびリソースタイプを呼び出すことができます。

```
root@zoneA:~# zonestat -z global -r physical-memory 2
```

注記 - zonestat を非大域ゾーンで使用すると、ほかのすべてのゾーン (大域ゾーンを含む) を合計したリソース使用状況が、大域ゾーンによって使用されたものとして報告されます。非大域ゾーンの zonestat ユーザーは、システムを共有しているほかのゾーンを認識しません。

▼ zonestat ユーティリティを使用して、CPU およびメモリーの使用効率のサマリーを表示する方法

1. root になるか、同等の役割になります。
2. CPU およびメモリー使用効率のサマリーを 5 秒ごとに表示します。

```
# zonestat -z global -r physical-memory 5
Collecting data for first interval...
Interval: 1, Duration: 0:00:05
PHYSICAL-MEMORY          SYSTEM MEMORY
mem_default              2046M
                          ZONE  USED %USED  CAP  %CAP
                          [total] 1020M 49.8%  -  -
                          [system] 782M 38.2%  -  -
                          global 185M 9.06%  -  -

Interval: 2, Duration: 0:00:10
PHYSICAL-MEMORY          SYSTEM MEMORY
mem_default              2046M
                          ZONE  USED %USED  CAP  %CAP
                          [total] 1020M 49.8%  -  -
                          [system] 782M 38.2%  -  -
                          global 185M 9.06%  -  -
...
```

▼ zonestat ユーティリティを使用して、デフォルトの pset について報告する方法

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. デフォルトの pset について 1 秒ごとに 1 分間報告します。

```
# zonestat -r default-pset 1 1m
Collecting data for first interval...
Interval: 1, Duration: 0:00:01
PROCESSOR_SET          TYPE  ONLINE/CPUS  MIN/MAX
pset_default          default-pset  2/2          1/-
                      ZONE  USED  PCT  CAP  %CAP  SHRS  %SHR  %SHRU
                      [total] 0.02 1.10%  -  -  -  -  -
                      [system] 0.00 0.19%  -  -  -  -  -
```

```

global 0.01 0.77% - - - -
zone1 0.00 0.07% - - - -
zone2 0.00 0.06% - - - -
...
Interval: 60, Duration: 0:01:00
PROCESSOR_SET
pset_default          TYPE ONLINE/CPUS      MIN/MAX
                      default-pset      2/2      1/-
                      ZONE USED PCT CAP %CAP SHRS %SHR %SHRU
[total] 0.06 3.26% - - - - -
[system] 0.00 0.18% - - - - -
  global 0.05 2.94% - - - - -
  zone1 0.00 0.06% - - - - -
  zone2 0.00 0.06% - - - - -

```

▼ zonestat を使用した合計および最大使用効率の報告

1. root になるか、同等の役割になります。
2. 表示なしで 10 秒ごとに 3 分間モニターしたあと、合計使用効率および最大使用効率の報告を生成します。

```

# zonestat -q -R total,high 10s 3m 3m
Report: Total Usage
  Start: Fri Aug 26 07:32:22 PDT 2011
  End: Fri Aug 26 07:35:22 PDT 2011
  Intervals: 18, Duration: 0:03:00
SUMMARY
  Cpus/Online: 2/2 PhysMem: 2046M VirtMem: 3069M
  ---CPU---- --PhysMem-- --VirtMem-- --PhysNet--
  ZONE USED %PART USED %USED USED %USED PBYTE %PUSE
[total] 0.01 0.62% 1020M 49.8% 1305M 42.5% 14 0.00%
[system] 0.00 0.23% 782M 38.2% 1061M 34.5% - -
  global 0.00 0.38% 185M 9.06% 208M 6.77% 0 0.00%
  test2 0.00 0.00% 52.4M 2.56% 36.6M 1.19% 0 0.00%

Report: High Usage
  Start: Fri Aug 26 07:32:22 PDT 2011
  End: Fri Aug 26 07:35:22 PDT 2011
  Intervals: 18, Duration: 0:03:00
SUMMARY
  Cpus/Online: 2/2 PhysMem: 2046M VirtMem: 3069M
  ---CPU---- --PhysMem-- --VirtMem-- --PhysNet--
  ZONE USED %PART USED %USED USED %USED PBYTE %PUSE
[total] 0.01 0.82% 1020M 49.8% 1305M 42.5% 2063 0.00%
[system] 0.00 0.26% 782M 38.2% 1061M 34.5% - -
  global 0.01 0.55% 185M 9.06% 207M 6.77% 0 0.00%
  test2 0.00 0.00% 52.4M 2.56% 36.6M 1.19% 0 0.00%

```

▼ 排他的 IP ゾーンでのネットワーク帯域幅の使用効率を取得する方法

zonestat コマンドに `-r` オプションと `network` リソースタイプを付けて実行すると、各ネットワークデバイスのゾーンごとの使用効率が表示されます。

各ゾーンによって使用された VNIC 形式のデータリンク帯域幅の量を表示するには、この手順を使用します。たとえば、net0 の下に表示されている zoneB は、このゾーンが VNIC 形式の net0 のリソースを消費したことを示しています。-x オプションを追加すると、特定の VNIC を表示できます。

1. 管理者になります。

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

2. -r オプションを付けた zonestat コマンドに network リソースタイプを指定して実行し、使用効率を一度に表示します。

```
# zonestat -r network 1 1
Collecting data for first interval...
Interval: 1, Duration: 0:00:01
```

NETWORK-DEVICE	SPEED			STATE	TYPE			
aggr1	2000mbps			up	AGGR			
	ZONE	TOBYTE	MAXBW	%MAXBW	PRBYTE	%PRBYTE	POBYTE	%POBYTE
	global	1196K	-	-	710K	0.28%	438K	0.18%
net0	1000mbps			up	PHYS			
	ZONE	TOBYTE	MAXBW	%MAXBW	PRBYTE	%PRBYTE	POBYTE	%POBYTE
	[total]	7672K	-	-	6112K	4.89%	1756K	1.40%
	global	5344K	100m*	42.6%	2414K	1.93%	1616K	1.40%
	zoneB	992K	100m	15.8%	1336K	0.76%	140K	0.13%
	zoneA	1336K	50m	10.6%	950K	1.07%	0	0.00%
net1	1000mbps			up	PHYS			
	ZONE	TOBYTE	MAXBW	%MAXBW	PRBYTE	%PRBYTE	POBYTE	%POBYTE
	global	126M	-	-	63M	6.30%	63M	6.30%
etherstub1	n/a			n/a	ETHERSTUB			
	ZONE	TOBYTE	MAXBW	%MAXBW	PRBYTE	%PRBYTE	POBYTE	%POBYTE
	[total]	3920K	-	-	0	-	0	-
	global	1960K	100M*	1.96%	0	-	0	-
	zoneA	1960K	50M	3.92%	0	-	0	-

すべてのゾーンのゾーン別の fstype 統計情報の報告

ゾーン別のファイルシステムアクティビティーについて報告するには、-z オプションを使用します。複数の -z オプションを使用して、選択したゾーンのアクティビティーをモニターできます。

指定した fstypes のファイルシステムアクティビティーのすべてのゾーンにわたる集計を報告するには、-A オプションを使用します。これは、-z または -Z オプションがどちらも使用されなかった場合のデフォルトの動作です。

-A オプションを -z または -Z オプションとともに使用すると、指定した fstypes のすべてのゾーンにわたる集計が別個の行に表示されます。

システム上のすべてのゾーンのファイルシステムアクティビティを報告するには、-z オプションを使用します。このオプションは、-z オプションとともに使用しても効果はありません。このオプションは、fstypes ではなく mountpoints のみをモニターする場合に使用しても効果はありません。

▼ -z オプションを使用して特定のゾーンのアクティビティをモニターする方法

- s10 ゾーンと s10u9 ゾーンのアクティビティをモニターするには、複数の -z オプションを使用します。

```
$ fsstat -z s10 -z s10u9 zfs tmpfs
new name name attr attr lookup rddir read read write write
file remov chng get set ops ops ops bytes ops bytes
 93 82 6 163K 110 507K 148 69.7K 67.9M 4.62K 13.7M zfs:s10
 248 237 158 188K 101 612K 283 70.6K 68.6M 4.71K 15.2M zfs:s10u9
12.0K 1.90K 10.1K 35.4K 12 60.3K 4 25.7K 29.8M 36.6K 31.0M tmpfs:s10
12.0K 1.90K 10.1K 35.6K 14 60.2K 2 28.4K 32.1M 36.5K 30.9M tmpfs:S10u9
```

▼ すべてのゾーンのゾーン別の fstype 統計情報を表示する方法

- システム上で実行されているゾーンごとに tmpfs および zfs ファイルシステムタイプのゾーン別の統計情報を取得し、tmpfs および zfs ファイルシステムタイプのシステム全体の集計も表示します。

```
$ fsstat -A -Z zfs tmpfs
new name name attr attr lookup rddir read read write write
file remov chng get set ops ops ops bytes ops bytes
360K 1.79K 20.2K 4.20M 1.02M 25.0M 145K 5.42M 2.00G 1.07M 8.10G zfs
359K 1.48K 20.1K 4.04M 1.02M 24.5M 144K 5.31M 1.88G 1.06M 8.08G zfs:global
 93 82 6 74.8K 107 250K 144 54.8K 60.5M 4.61K 13.7M zfs:s10
 248 237 158 90.2K 101 336K 283 53.0K 58.3M 4.71K 15.2M zfs:s10u9
60.0K 41.9K 17.7K 410K 515 216K 426 1022K 1.02G 343K 330M tmpfs
49.4K 38.1K 11.0K 366K 489 172K 420 968K 979M 283K 273M tmpfs:global
5.28K 1.90K 3.36K 21.9K 12 21.7K 4 25.7K 29.8M 29.9K 28.3M tmpfs:s10
5.25K 1.90K 3.34K 22.1K 14 21.6K 2 28.4K 32.1M 29.8K 28.2M tmpfs:s10u9
```

この出力では、システム上の非大域ゾーンは s10 と s10u9 です。

非大域ゾーン内での DTrace の使用

147 ページの「非大域ゾーン内での DTrace の実行」で説明されている DTrace 機能を使用するには、次の手順を実行します。

▼ DTrace を使用する方法

1. `zonecfg limitpriv` プロパティを使用して、`dtrace_proc` 特権と `dtrace_user` 特権を追加します。

```
global# zonecfg -z my-zone
zonecfg:my-zone> set limitpriv="default,dtrace_proc,dtrace_user"
zonecfg:my-zone> exit
```

注記 - 必要に応じて、どちらか一方の特権を追加することも、両方の特権を追加することもできます。

2. ゾーンをブートします。
3. ゾーンにログインします。
4. DTrace プログラムを実行します。

```
global# zoneadm -z my-zone boot
```

```
global# zlogin my-zone
```

```
my-zone# dtrace -l
```

非大域ゾーンの SMF サービスのステータスの確認

非大域ゾーンの SMF サービスのステータスを確認するには、`zlogin` コマンドを使用します。

▼ コマンド行から SMF サービスのステータスを確認する方法

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. コマンド行から次のように入力して、無効になっているサービスも含むすべてのサービスを表示します。

```
global# zlogin my-zone svcs -a
```

参照 詳細については、第5章「非大域ゾーンへのログイン」および `svcs(1)` を参照してください。

▼ ゾーン内から SMF サービスのステータスを確認する方法

1. 管理者になります。

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

2. ゾーンにログインします。

```
global# zlogin my-zone
```

3. `svcs` コマンドに `-a` オプションを付けて実行して、無効になっているサービスも含むすべてのサービスを表示します。

```
my-zone# svcs -a
```

参照 詳細については、第5章「非大域ゾーンへのログイン」および `svcs(1)` を参照してください。

稼働中の非大域ゾーン内でファイルシステムをマウントする

稼働中の非大域ゾーン内でファイルシステムをマウントできます。具体的には、次の作業について説明しています。

- 大域管理者または大域ゾーン内で適切な承認を付与されたユーザーとして、`raw` デバイスおよびブロックデバイスを非大域ゾーンにインポートできます。デバイスのインポート後に、ゾーン管理者はディスクにアクセスできます。その後、ゾーン管理者はディスク上に新しいファイルシステムを作成して、次のいずれかの操作を実行できます。
 - ファイルシステムを手動でマウントします
 - ファイルシステムがゾーンのブート時にマウントされるように、`/etc/vfstab` 内に配置します
- 大域管理者または適切な承認を付与されたユーザーとして、大域ゾーンから非大域ゾーンにファイルシステムをマウントすることもできます。

大域ゾーンから非大域ゾーンにファイルシステムをマウントする前に、非大域ゾーンは準備完了状態であるか、ブートされている必要があります。それ以外の場合、次回ゾーンをブートまたは準備完了状態にしようとする失敗します。また、大域ゾーンから非大域ゾーンにマウントされたファイルシステムは、ゾーンが停止されるときにアンマウントされます。

▼ LOFS を使用してファイルシステムをマウントする方法

LOFS マウントを使用すると、大域ゾーンと非大域ゾーンの間でファイルシステムを共有できます。この手順では、`zonecfg` コマンドを使用して、大域ゾーンの `/export/datafiles` ファイルシステムの LOFS マウントを `my-zone` 構成に追加します。この例では、マウントオプションをカスタマイズしません。

この手順を実行するには、大域管理者またはゾーンセキュリティ権利プロファイルを持つ大域ゾーンのユーザーである必要があります。

1. 管理者になります。

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

2. `zonecfg` コマンドを使用します。

```
global# zonecfg -z my-zone
```

3. ファイルシステムを構成に追加します。

```
zonecfg:my-zone> add fs
```

4. ファイルシステムのマウントポイントに `my-zone` の `/datafiles` を設定します。

```
zonecfg:my-zone:fs> set dir=/datafiles
```

5. 大域ゾーンの `/export/datafiles` が、`my-zone` で `/datafiles` としてマウントされるように指定します。

```
zonecfg:my-zone:fs> set special=/export/datafiles
```

6. ファイルシステムのタイプを設定します。

```
zonecfg:my-zone:fs> set type=lofs
```

7. 指定を終了します。

```
zonecfg:my-zone:fs> end
```

8. 構成を検証および確定します。

```
zonecfg:my-zone> verify
zonecfg:my-zone> commit
```

▼ ZFS データセットを非大域ゾーンに委任する方法

ZFS データセットを非大域ゾーンに委任するには、この手順を使用します。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を付与されたユーザーである必要があります。

1. 管理者になります。

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

2. 大域ゾーンから、pool1A という既存の ZFS プールに fs2 という新しい ZFS ファイルシステムを作成します。

```
global# zfs create pool1A/fs2
```

3. (オプション) pool1A/fs2 ファイルシステムの mountpoint プロパティに /fs-del/fs2 を設定します。

```
global# zfs set mountpoint=/fs-del/fs2 pool1A/fs2
```

mountpoint を設定する必要はありません。mountpoint プロパティを指定しない場合、データセットはデフォルトでゾーン内の /alias にマウントされます。mountpoint および canmount プロパティをデフォルト以外の値にすると、zfs(1M) のマニュアルページで説明されているように、この動作が変更されます。

4. このファイルシステムの mountpoint プロパティのソースが、現在 local であることを確認します。

```
global# zfs get mountpoint pool1A/fs2
NAME          PROPERTY      VALUE          SOURCE
pool1A/fs2    mountpoint    /fs-del/fs2    local
```

5. pool1A/fs2 ファイルシステムを委任するか、別名のデータセットを指定します。

■ pool1A/fs2 ファイルシステムをゾーンに委任します。

```
# zonecfg -z my-zone
zonecfg:my-zone> add dataset
zonecfg:my-zone:dataset> set name=pool1A/fs2
zonecfg:my-zone:dataset> end
```

■ 別名のデータセットを指定します。

```
# zonecfg -z my-zone
zonecfg:my-zone> add dataset
zonecfg:my-zone:dataset> set name=pool1A/fs2
zonecfg:my-zone:dataset> set alias=delegated
zonecfg:my-zone:dataset> end
```

6. ゾーンをリブートして、すべての pool1A ファイルシステムの zoned プロパティを表示します。

```
global# zfs get -r zoned pool1A
```

NAME	PROPERTY	VALUE	SOURCE
poolA	zoned	off	default
poolA/fs2	zoned	on	default

poolA/fs2 の zoned プロパティが on に設定されます。この ZFS ファイルシステムは、非大域ゾーンに委任されてゾーンにマウントされ、ゾーン管理者の制御下にあります。ZFS は、zoned プロパティを使用して、データセットがある時点で非大域ゾーンに委任されたことを示します。

大域ゾーン内の特定のファイルシステムへのアクセス権を非大域ゾーンに追加する

▼ 非大域ゾーンで CD または DVD メディアにアクセスする権限を追加する方法

この手順を使用して、非大域ゾーンで CD または DVD メディアに読み取り専用のアクセスを行う権限を追加できます。ループバックファイルシステム lofs は、CD または DVD のコンテンツを非大域ゾーンで利用できるようにするために使用されます。このメディアは非大域ゾーンにマウントされます。メディアをマウントする際は、大域ゾーンでボリューム管理ファイルシステムが使用されます。アクセス権を追加したら、CD または DVD を使用して製品を非大域ゾーンにインストールできます。この手順では、jes_05q4_dvd という DVD を使用します。

1. 管理者になります。

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

2. メディアを挿入します。

3. ドライブにメディアが入っているかどうかを確認します。

```
global# volcheck
```

4. DVD が自動マウントされているかどうかをテストします。

```
global# ls /cdrom
```

次のような情報が表示されます。

```
cdrom  cdrom1  jes_05q4_dvd
```

5. **ro,nodevices** オプション (読み取り専用、デバイスなし) を指定して、非大域ゾーンでファイルシステムをループバックマウントします。

```
global# zonecfg -z my-zone
zonecfg:my-zone> add fs
zonecfg:my-zone:fs> set dir=/cdrom
zonecfg:my-zone:fs> set special=/cdrom
zonecfg:my-zone:fs> set type=lofs
zonecfg:my-zone:fs> add options [ro,nodevices]
zonecfg:my-zone:fs> end
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

6. 非大域ゾーンをリブートします。

```
global# zoneadm -z my-zone reboot
```

7. zoneadm list コマンドに -v オプションを指定して、ステータスを確認します。

```
global# zoneadm list -v
```

次のような情報が表示されます。

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	my-zone	running	/zones/my-zone	solaris	excl

8. 非大域ゾーンにログインします。

```
global# my-zone
```

9. DVD-ROM がマウントされているかを確認します。

```
my-zone# ls /cdrom
```

次のような内容が表示されます。

```
cdrom  cdrom1  jes_05q4_dvd
```

10. 製品のインストールガイドで説明されているとおりに、製品をインストールします。

11. 非大域ゾーンから抜けます。

```
my-zone# exit
```

ヒント - /cdrom ファイルシステムを非大域ゾーンに残すこともできます。マウントするときには、常に CD-ROM ドライブの現在の内容が反映されます。つまり、ドライブが空の場合は、ディレクトリは空になります。

12. (オプション) 非大域ゾーンから /cdrom ファイルシステムを削除する場合は、次の手順を使用します。

```
global# zonecfg -z my-zone
zonecfg:my-zone> remove fs dir=/cdrom
zonecfg:my-zone> commit
zonecfg:my-zone> exit
```

ゾーンがインストールされている Oracle Solaris システムでの IP ネットワークマルチパスの使用

▼ 排他的 IP 非大域ゾーンで IP ネットワークマルチパスを使用する方法

排他的 IP ゾーンの IP ネットワークマルチパス (IPMP) は、大域ゾーンと同じ方法で構成します。IPMP を使用するには、排他的 IP ゾーンに少なくとも 2 つの `zonecfg add net` リソースが必要です。IPMP は、これらのデータリンク上のゾーン内から構成されます。

IP マルチパスグループ (IPMP グループ) に 1 つ以上の物理インタフェースを構成できます。IPMP を構成すると、IPMP グループのインタフェースに障害が発生していないかどうかをシステムが自動的にモニターします。グループのインタフェースに障害が発生した場合や、保守のためにインタフェースが削除された場合、IPMP は自動的に、そのインタフェースの IP アドレスを移行して処理を継続します。フェイルオーバーされたアドレスは、障害が発生したインタフェースの IPMP グループ内の機能中のインタフェースが受け取ります。IPMP のフェイルオーバーコンポーネントは、接続を保持し、既存の接続の切断を防止します。さらに、IPMP は、ネットワークトラフィックを自動的に IPMP グループ内のインタフェースのセットに分散することによって、ネットワークパフォーマンス全体を向上させます。この処理は負荷分散と呼ばれます。

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. 『Oracle Solaris 11.3 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』の「IPMP グループの構成」で説明されているように、IPMP グループを構成します。

▼ IP ネットワークマルチパス機能を共有 IP 非大域ゾーンに拡張する方法

大域ゾーン内で IPMP を構成し、IPMP 機能を非大域ゾーンに拡張する場合に、ここで説明する手順を使用します。

ゾーンの構成時に、各アドレスつまり論理インタフェースを非大域ゾーンと関連付ける必要があります。手順については、『Oracle Solaris ゾーン構成リソース』の

「[zonecfg コマンドの使用について](#)」および [22 ページの「ゾーンの構成方法](#)」を参照してください。

この手順を実行すると、次のことが達成されます。

- net0 カードと net1 カードは、インタフェースが `ipmp0` である IPMP グループと一緒に構成されます。
- `ipmp0` のアドレス `192.168.0.1` が非大域ゾーン `my-zone` と関連付けられます。
- net0 カードがゾーンの物理インタフェースとして設定されます。

稼働中のゾーンで、`ipadm` コマンドを使用して関連付けを行うことができます。詳細は、[132 ページの「共有 IP ネットワークインタフェース](#)」および `ipadm(1M)` のマニュアルページを参照してください。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を付与されたユーザーである必要があります。

1. 管理者になります。

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

2. 大域ゾーンで、『[Oracle Solaris 11.3 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理](#)』の「[IPMP グループの構成](#)」の説明に従って IPMP グループを構成します。

3. `zonecfg` コマンドを使用してゾーンを構成します。net リソースを構成する際、アドレス `192.168.0.1` および物理インタフェース `net0` をゾーン `my-zone` に追加します。

```
zonecfg:my-zone> add net
zonecfg:my-zone:net> set address=192.168.0.1
zonecfg:my-zone:net> set physical=net0
zonecfg:my-zone:net> end
```

非大域ゾーン `my-zone` 内に `net0` だけが表示されます。

net0 に連続して障害が発生した場合

`192.168.0.1` が `ipmp0` に割り当てられているので、`net0` に連続して障害が発生した場合、このアドレスは引き続き `net1` から利用できます。アドレスの可用性はまた、`my-zone` にも適用されます。ただし、`net0` に障害が発生した場合、`net1` が `my-zone` の可視インタフェースになります。

排他的 IP 非大域ゾーンでのデータリンクの管理

データリンクを管理するには、大域ゾーンから `dladm` コマンドを使用します。

▼ dladm show-linkprop の使用方法

dladm コマンドを show-linkprop サブコマンドとともに使用すると、実行中の排他的 IP ゾーンに対するデータリンクの割り当てを表示できます。

データリンクを管理するには、大域管理者または適切な承認を付与された大域ゾーンのユーザーである必要があります。

1. 管理者になります。

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

2. システムのデータリンクの割り当てを表示します。

```
global# dladm show-linkprop
```

例 8 ゾーンへのデータリンク割り当ての表示

```
1. global# dladm show-linkprop
```

```
2. net0 リンクはゾーン vz1-100 に割り当てられています。
```

```
global# dladm show-linkprop
LINK      PROPERTY      PERM VALUE      EFFECTIVE    DEFAULT    POSSIBLE
net2      speed         r-  10         10          10         --
...
vz1-100/net0 autopush  rw  --         --          --         --
vz1-100/net0 zone      rw  vz1-100     vz1-100     --         --
vz1-100/net0 state     r-  up         up          up         up,down
vz1-100/net0 mtu       rw  1500       1500       1500      576-1500
vz1-100/net0 maxbw     rw  --         --          --         --
vz1-100/net0 cpus      rw  --         0-3,8-11   --         --
vz1-100/net0 rxfanout  rw  --         8          1          --
vz1-100/net0 pool      rw  --         --          --         --
vz1-100/net0 priority  rw  high       high       high      low,medium,
                                         high
vz1-100/net0 tagmode   rw  vlanonly   vlanonly   vlanonly  normal,
                                         vlanonly
vz1-100/net0 protection rw  mac-nospoof mac-nospoof --      mac-nospoof,
                                         restricted,
                                         ip-nospoof,
                                         dhcp-nospoof
vz1-100/net0 mac-address rw  0:16:3e:86:11:f5 0:16:3e:86:11:f5 0:16:3e:86:11:f5 --
vz1-100/net0 allowed-ips rw  --         --          --         --
vz1-100/net0 allowed-dhcp-cids rw --         --          --         --
vz1-100/net0 rxrings   rw  --         --          --         --
```

```

vz1-100/net0 txrings      rw  --          --          --          sw, hw
vz1-100/net0 txringsavail r-  0          0          --          --
vz1-100/net0 rxringsavail r-  0          0          --          --
vz1-100/net0 rxhwclntavail r-  0          0          --          --
vz1-100/net0 txhwclntavail r-  0          0          --          --
vz1-100/net0 vsi-typeid  rw  --          116         --          --
vz1-100/net0 vsi-vers    rw  --          0           --          --
vz1-100/net0 vsi-mgrid   rw  --          ::          --          --
vz1-100/net0 vsi-mgrid-enc rw --          oracle_v1   oracle_v1 none,
                                                oracle_v1
vz1-100/net0 lro         rw  off         off         auto        on, off, auto
vz1-100/net0 cos         rw  --          --          0           --
vz1-100/net0 etsbw-lcl   rw  --          --          0           --
vz1-100/net0 etsbw-rmt   r-  --          --          --          --
vz1-100/net0 etsbw-lcl-advice r- --          --          --          --
vz1-100/net0 etsbw-rmt-advice rw --          --          0           --

```

例 9 バニティーネーミングを使用している場合に、データリンク名および物理的な場所を表示する方法

デバイスの物理的な場所は、LOCATION フィールドに表示されます。デバイスのデータリンク名および物理的な場所の情報を表示するには、-L オプションを使用します。

```

global# dladm show-phys -L
LINK          DEVICE      LOCATION
net0          net0        MB
net1          net1        MB
net2          net2        MB
net3          net3        MB
net4          ibp0        MB/RISER0/PCIE0/PORT1
net5          ibp1        MB/RISER0/PCIE0/PORT2
net6          eoib2        MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
net7          eoib4        MB/RISER0/PCIE0/PORT2/cloud-nm2gw-2/1A-ETH-2

```

▼ dladm を使用して、一時的なデータリンクを割り当てる方法

dladm コマンドを set-linkprop サブコマンドとともに使用すると、実行中の排他的 IP ゾーンに対してデータリンクを一時的に割り当てることができます。持続的な割り当ては zonecfg コマンドを使用して行う必要があります。

データリンクを管理するには、大域管理者または適切な承認を付与された大域ゾーンの利用者である必要があります。

1. 管理者になります。

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

2. **dladm set-linkprop** を **-t** とともに使用して、**zoneA** という実行中のゾーンに **net0** を追加します。

```
global# dladm set-linkprop -t -p zone=zoneA net0
LINK      PROPERTY      PERM  VALUE      DEFAULT  POSSIBLE
net0      zone           rw    zoneA      --       --
```

ヒント **--p** オプションを使用すると、マシンで構文解析できる安定した形式の表示が生成されます。

▼ dladm reset-linkprop の使用方法

dladm コマンドを **reset-linkprop** サブコマンドとともに使用すると、**net0** リンクの値を未設定にリセットできます。

1. 管理者になります。
詳細は、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティー保護』の「割り当てられている管理権利の使用」を参照してください。
2. **dladm reset-linkprop** を **-t** オプションとともに使用して、ゾーンへの **net0** デバイスの割り当てを解除します。

```
global# dladm reset-linkprop -t -p zone net0
LINK      PROPERTY      PERM  VALUE      DEFAULT  POSSIBLE
net0      zone           rw    zoneA      --       --
```

ヒント **--p** オプションを使用すると、マシンで構文解析できる安定した形式の表示が生成されます。

注意事項 実行中のゾーンでそのデバイスが使用中の場合、再割り当ては失敗し、エラーメッセージが表示されます。185 ページの「排他的 IP ゾーンがデバイスを使用しているために **dladm reset-linkprop** が失敗する」を参照してください。

ゾーンがインストールされている Oracle Solaris システムでの公平配分スケジューラの使用

prctl コマンドで指定された制限には持続性がありません。システムがリブートされると、制限は無効になります。ゾーンの配分を持続的に設定する方法について

は、22 ページの「ゾーンの構成方法」および 32 ページの「大域ゾーンの `zone.cpu-shares` を設定する方法」を参照してください。

▼ prctl コマンドを使用して大域ゾーンの FSS 配分を設定する方法

大域ゾーンには、デフォルトで 1 つの配分が付与されます。ここで説明する手順を使って、デフォルトの割り当てを変更できます。システムをリブートするたびに `prctl` コマンドで割り当てた配分を設定し直す必要があります。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を付与されたユーザーである必要があります。

1. 管理者になります。

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

2. `prctl` ユーティリティーを使用して、2 つの配分を大域ゾーンに割り当てます。

```
# prctl -n zone.cpu-shares -v 2 -r -i zone global
```

3. (オプション) 大域ゾーンに割り当てられた配分の数を確認するには、次のように入力します。

```
# prctl -n zone.cpu-shares -i zone global
```

参照 `prctl` ユーティリティーの詳細は、[prctl\(1\)](#) のマニュアルページを参照してください。

▼ ゾーンの `zone.cpu-shares` 値を動的に変更する方法

この手順は、大域ゾーンまたは非大域ゾーンで使用できます。

1. 管理者になります。

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

2. `prctl` コマンドを使って `cpu-shares` の新しい値を指定します。

```
# prctl -n zone.cpu-shares -r -v value -i zone zonename
```

`idtype` は `zonename`、`zoneid` のいずれかです。 `value` は新しい値です。

ゾーン管理での権利プロファイルの使用

このセクションでは、非大域ゾーンで権利プロファイルを使用することに関連したタスクについて説明します。

▼ Zone Management プロファイルを割り当てる方法

ユーザーは、Zone Management プロファイルを使用することで、システムの非大域ゾーンをすべて管理できます。

この手順を実行するには、大域管理者または大域ゾーン内で適切な承認を付与されたユーザーである必要があります。

1. 管理者になります。

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

2. Zone Management 権利プロファイルを含む役割を作成し、その役割をユーザーに割り当てます。

ゾーンがインストールされている Oracle Solaris システムのバックアップ

次の手順を使用して、ゾーン内のファイルをバックアップできます。必ずゾーンの構成ファイルもバックアップしてください。

▼ 統合アーカイブを使用した Oracle Solaris システムのバックアップ

1. 管理者になります。

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

2. 復旧用アーカイブを作成します。

```
source-host# archiveadm create -r -z zonename archive-name
```

たとえば、ゾーン名 `my-zone` とアーカイブパス `/net/server/my-zone-archive.uar` を使用して、次のように指定します。

```
host1# archiveadm create -r -z my-zone /net/server/my-zone-archive.uar
```

▼ x64: ゾーン構成のコピーを出力する方法

非大域ゾーン構成のバックアップファイルを作成することをお勧めします。必要に応じて、このバックアップを使用してゾーンをあとで再作成できます。ゾーンにはじめてログインして `sysidtool` からの質問に回答したあとに、ゾーンの構成のコピーを作成します。この手順では、`my-zone` という名前のゾーンと `my-zone.config` という名前のバックアップファイルを使用します。

1. 管理者になります。

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

2. ゾーン `my-zone` の構成を `my-zone.config` というファイルに出力します。

```
global# zonecfg -z my-zone export > my-zone.config
```

非大域ゾーンの再作成

▼ 非大域ゾーンを個別に再作成する方法

必要に応じて、非大域ゾーン構成のバックアップファイルを使用して非大域ゾーンを再作成できます。この手順では、ゾーンを再作成する処理をわかりやすく説明するために、`my-zone` という名前のゾーンと `my-zone.config` という名前のバックアップファイルを使用します。

1. 管理者になります。

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

2. ゾーン `my-zone` を再作成するために、`my-zone.config` を `zonecfg` コマンドファイルとして使用することを指定します。

```
global# zonecfg -z my-zone -f my-zone.config
```

3. ゾーンをインストールします。

```
global# zoneadm -z my-zone install -a /path/to/archive options
```

4. ゾーン固有のファイル (アプリケーションデータなど) を復元する場合は、新しく作成したゾーンのルートファイルシステムに対してバックアップのファイルを手動で復元します。手作業でのマージが必要な場合があります。

不変ゾーンの構成と管理

不変ゾーンは、非大域ゾーンおよび大域ゾーンに対して読み取り専用ファイルシステムプロファイルを提供します。物理グローバルゾーンとカーネルゾーン (これは仮想グローバルゾーンです) の両方とも不変ゾーンになることができます。

読み取り専用ゾーンの概要

読み取り専用ゾーンルートを持つゾーンは、不変ゾーンと呼ばれます。不変ゾーンは、ゾーンに読み取り専用の `root` ファイルシステムを実装することにより、ゾーンの構成を維持します。このゾーンでは、実行時環境に追加の制限を加えることにより、ゾーンの安全な実行時の境界が拡張されます。特定の保守操作として実行される場合を除き、システムバイナリまたはシステム構成への変更はブロックされます。

必須書き込みアクセス制御 (MWAC) カーネルポリシーを使用すると、`zoncfg file-mac-profile` プロパティを介して、ファイルシステム書き込み権限が適用されます。大域ゾーンは MWAC ポリシーの対象ではないため、大域ゾーンはインストール、イメージの更新、および保守のために非大域ゾーンのファイルシステムに書き込むことができます。

ゾーンが準備完了状態になると、MWAC ポリシーがダウンロードされます。このポリシーはゾーンのブート時に有効になります。インストール後のアセンブリと構成を行うには、一時的な書き込み可能ルートファイルシステムのブート処理を使用します。ゾーンの MWAC 構成への変更は、ゾーンのリブートによってのみ有効になります。

ゾーンの構成、インストール、およびブートの一般的な情報については、[第1章「非大域ゾーンの計画および構成方法」](#) および [第3章「非大域ゾーンのインストール、ブート、停止処理、停止、アンインストール、およびクローニング」](#) を参照してください。

読み取り専用ゾーンの構成

zonecfg の file-mac profile プロパティ

デフォルトでは、zonecfg file-mac-profile プロパティは非大域ゾーンに設定されません。ゾーンは、書き込み可能なルートデータセットを持つように構成されません。

solaris 読み取り専用ゾーンでは、読み取り専用ゾーンルートを構成するために、file-mac-profile プロパティが使用されます。読み取り専用ルートは、ゾーン内からの実行時環境へのアクセスを制限します。

zonecfg ユーティリティを使用すると、file-mac-profile を次のいずれかの値に設定できます。none 以外のすべてのプロファイルでは、/var/pkg ディレクトリとその内容がゾーンの内部から読み取り専用になります。

none	既存のゾーン境界のほかに追加の保護がない標準の読み取り書き込み非大域ゾーン。値を none に設定することは、file-mac-profile プロパティを設定しないことと同等になります。
strict	読み取り専用ファイルシステム。例外はありません。 <ul style="list-style-type: none"> ■ IPS パッケージはインストールできません。 ■ 持続的に有効な SMF サービスは固定されます。 ■ SMF マニフェストはデフォルトの場所から追加できません。 ■ 構成ファイルのロギングおよび監査は固定されます。データはリモートからのみロギングできます。 ■ このプロファイルで大域不変ゾーンまたは非大域不変ゾーン内の NFS サーバーの実行はサポートされていません。fixed-configuration プロファイルを使用して NFS サーバーを実行する必要があります。
fixed-configuration	システム構成コンポーネントが含まれているディレクトリを除く、/var/* ディレクトリへの更新を許可します。 <ul style="list-style-type: none"> ■ IPS パッケージ (新しいパッケージを含む) はインストールできません。 ■ 持続的に有効な SMF サービスは固定されます。 ■ SMF マニフェストはデフォルトの場所から追加できません。 ■ 構成ファイルのロギングおよび監査はローカルに行うことができます。syslog および監査の構成は固定されます。
flexible-configuration	/etc/* ディレクトリ内のファイルの変更、ルートのホームディレクトリの変更、および /var/* ディレクトリへの更新を許可しま

す。この構成は、Oracle Solaris 10 のネイティブな疎ルートゾーン (『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』を参照) に近い機能を提供します。これは、このガイドの Oracle Solaris 10 バージョンです。

- IPS パッケージ (新しいパッケージを含む) はインストールできません。
- 持続的に有効な SMF サービスは固定されます。
- SMF マニフェストはデフォルトの場所から追加できません。
- 構成ファイルのロギングおよび監査はローカルに行うことができます。syslog および監査の構成は変更できます。

dynamic-zones

dynamic-zones 構成は、カーネルゾーンと非大域ゾーンを作成および破棄する機能を提供します。このプロファイルは、大域ゾーン (カーネルゾーンの大域ゾーンを含む) に対して有効です。

dynamic-zones 構成は fixed-configuration と同等ですが、カーネルゾーンと非大域ゾーンを作成および破棄する機能を追加します。dynamic-zones 構成は flexible-configuration と同等ですが、flexible-configuration では /etc 内のファイルへの書き込みも許可されます。

例 10 プロファイルの設定

プロファイルを選択してアクティブ化するには、次のコマンドを使用します。この例では、dynamic-zones プロファイルを使用します。

```
# zonecfg -z global set file-mac-profile=dynamic-zones
```

zonecfg add dataset リソースポリシー

add dataset リソースを使用してゾーンに追加されたデータセットは、MWAC ポリシーの対象にはなりません。追加のデータセットを委任されたゾーンは、それらのデータセットを完全に制御できます。プラットフォームのデータセットは表示されますが、ゾーンが読み取り/書き込みでブートされないかぎり、そのデータとプロパティは読み取り専用です。

zonecfg add fs リソースポリシー

add fs リソースを使用してゾーンに追加されたファイルシステムは、MWAC ポリシーの対象にはなりません。ファイルシステムは読み取り専用でマウントできます。

読み取り専用ゾーンの管理

ディスク上の構成は大域ゾーンからのみ管理できます。実行中のゾーン内では、ゾーンが書き込み可能でブートされた場合を除き、管理は実行時状態の設定に制限されず。SMF コマンド (`svcadm(1M)` および `svccfg(1M)` のマニュアルページに記載) を使用して行われた構成変更は、ディスク上の SMF データベースではなく、一時的な使用中の SMF データベースにのみ適用できます。ゾーンの MWAC 構成への変更は、ゾーンのリブート時に有効になります。

初期インストールまたはアップデートのあと、`self-assembly-complete` マイルストーンに達するまで、ゾーンは一時的な読み取り/書き込みでブートされます。その後、ゾーンは読み取り専用モードでリブートされます。

zoneadm list -p の表示

解析可能な出力に、R/W 列および file-mac-profile 列が表示されます。

```
global# zoneadm list -p
0:global:running:/:UUID:solaris:shared:-:none
5:testzone2:running:/export/zones/testzone2:UUID \
:solaris:shared:R:fixed-configuration
12:testzone3:running:/export/zones/testzone3:UUID \
:solaris:shared:R:fixed-configuration
13:testzone1:running:/export/zones/testzone1:UUID \
:solaris:excl:W:fixed-configuration
-:testzone:installed:/export/zones/testzone:UUID \
:solaris:excl:-:fixed-configuration
```

R および W オプションは、次のように定義されます。

- -R は、読み取り専用でブートされる file-mac-profile を持つゾーンを示します。
- -W は、読み取り/書き込みでブートされる file-mac-profile を持つゾーンを示します。

書き込み可能なルートファイルシステムを持つ読み取り専用ゾーンをブートするオプション

`zoneadm boot` サブコマンドを使用すると、書き込み可能なルートファイルシステムまたは一時的に書き込み可能なルートファイルシステムのいずれかを持つ読み取り専用ゾーンを手動でブートできる 2 つのオプションが大域ゾーン管理者に提供されます。ゾーンは次にリブートが行われるまで書き込み可能モードとなります。

- w 書き込み可能なルートファイルシステムを持つゾーンを手動でブートします。
 - W 一時的に書き込み可能なルートファイルシステムを持つゾーンを手動でブートします。システムは、`self-assembly-complete` マイルストーンに達すると自動的にリブートされます。
リブートすると、ゾーンがふたたび MWAC ポリシーの制御下に置かれます。このオプションは、ゾーンが MWAC ポリシー `none` を持っている場合に許可されます。
- w および -W オプションは、非不変ゾーンではないゾーンでは無視されます。

zlogin コマンドを使用したファイルの編集またはパッケージの追加

`zlogin` コマンドは、不変ファイルの編集や新しいパッケージの追加などのアクションに対し 2 つのオプションを提供します。これらのオプションを使用するには、`solaris.zone.manage/zonename` の承認が必要です。

- T トラステッドパス属性 `PRIV_PROC_TPD` を設定して不変ゾーンに入ります。このセッションでは、通常は変更できないゾーン内のファイルを変更できます。保護されていないファイルはセッションで読み取ることはできません。
- U `-T` オプションと同じ処理を実行しますが、安全でないモードで行います。このオプションは、`file-mac-profile` プロパティの `flexible-configuration` が設定されたゾーンで必要です。

注記 - これらのオプションは、コンソールログインでは使用できません。どちらのオプションも、不変ゾーンではないゾーンの場合、無視されます。

不変大域ゾーン

不変大域ゾーンは、不変非大域ゾーンを大域ゾーンに拡張します。

不変大域ゾーンの構成

大域ゾーンの構成は、`zonecfg` コマンド `file-mac-profile` を通じて行われます。使用可能な値は、[180 ページの「zonecfg の file-mac profile プロパティ」](#) に記

載されています。システムで DHCP を使用してネットワークインタフェースを設定している場合、`flexible-configuration` を選択する必要があります。

`rpool` データセットは制限されますが、`add dataset` を使用することにより、無制限のサブデータセットを追加できます。不変大域ゾーンは、無制限のデータセットでのみゾーンを実行できます。無制限のデータセットのすべての子も無制限です。

ゾーン構成を確定したあと、`zonecfg` ブート情報が書き込まれ、ブートアーカイブが更新されます。ゾーンがすぐに不変になります。リブートは必要ありません。

不変大域ゾーンの保守

トラステッドパスアクセスを使用して大域ゾーンを管理します。トラステッドパスはコンソールでのみ使用可能なので、ILOM、シリアル接続、またはグラフィカルコンソールを通じてコンソールにアクセスできることを確認してください。

不変大域ゾーンとしてシステムを構成したら、トラステッドパスコンソールにアクセスします。ログインし、`root` 役割になります。

パッケージ更新を実行した場合、不変大域ゾーンの最初のブートは読み取り書き込みになります。必要な自己アセンブリ手順を実行するために、これらのアクセス権がシステムが必要です。自己アセンブリ手順の実行が終わると、システムは再度不変になります。

トラステッドパスについては、『[Trusted Extensions 構成と管理](#)』を参照してください。

Oracle Solaris ゾーンで発生するさまざまな問題のトラブルシューティング

この章では、ゾーンのトラブルシューティングについて説明します。

排他的 IP ゾーンがデバイスを使用しているために `dladm reset-linkprop` が失敗する

次のエラーメッセージが表示された場合

```
dladm: warning: cannot reset link property 'zone' on 'net0': operation failed
```

174 ページの「[dladm reset-linkprop の使用方法](#)」で説明されているように、`dladm reset-linkprop` を使用しようとして失敗しました。実行中のゾーン `exc1` がそのデバイスを使用しています。

値をリセットするには、次の手順に従います。

1. 次のように入力します。

```
global#ipadm delete-ip net0
```

2. `dladm` コマンドを再実行します。

ゾーン構成内に不正な特権セットが指定されている

ゾーンの特権セットに許可されない特権が含まれる場合、必須の特権が欠落している場合、または不明な特権名が含まれる場合、ゾーンの検証、準備、またはブートの試行は失敗し、次のようなエラーメッセージが表示されます。

```
zonecfg:zone5> set limitpriv="basic"
.
.
global# zoneadm -z zone5 boot
required privilege "sys_mount" is missing from the zone's privilege set
```

```
zoneadm: zone zone5 failed to verify
```

ゾーンが停止しない

ゾーンに関連付けられたシステム状態を破棄できない場合には、停止処理は途中で失敗します。このため、稼働中とインストール済みの間で、ゾーンが中間状態のままになります。この状態では、アクティブなユーザープロセスやカーネルスレッドは存在せず、何も作成できません。停止操作が失敗した場合は、手動で処理を完了する必要があります。

障害のもっとも一般的な原因は、システムがすべてのファイルシステムをアンマウントできないことです。システム状態が破棄される Oracle Solaris システムの従来の停止処理とは異なり、ゾーンでは、ゾーンのブートまたは操作時に実行されたマウントがゾーンの停止後に残らないことを保証する必要があります。zoneadm によりゾーン内で実行中のプロセスが存在しないことが確認されても、大域ゾーン内のプロセスによりゾーン内のファイルが開かれた場合には、アンマウント操作が失敗することがあります。proc(1) (pfiles に記載) および fuser(1M) のマニュアルページに記載されているツールを使用してこれらのプロセスを検索し、適切なアクションを実行してください。これらのプロセスを処理したあとで zoneadm halt を再度呼び出すと、ゾーンが完全に停止します。

共有ストレージでの Oracle Solaris ゾーンの使用開始

Oracle Solaris の共有ストレージ機能で Oracle Solaris ゾーンを使用すると、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および管理できます。これらの自動化された機能により、Oracle Solaris システムでのゾーンとそれに対応する共有ストレージリソースの配備、管理、および移行が簡単になります。

対応する共有ストレージリソースについて、ホストに依存しない形式でゾーン構成に記述できます。この機能を使用したゾーンインストールは、共有ストレージデバイスでホストされる専用の ZFS ストレージプールにカプセル化されます。

Oracle Solaris ゾーン フレームワークは自動的に、共有ストレージリソースを構成および構成解除します。さまざまなゾーン管理アクティビティー全体で必要になる ZFS ストレージプール管理タスクがすべて自動的に実行されます。

ストレージ URI を使用した共有ストレージリソースについて

ストレージ URI は、ホストに依存しない形式で共有ストレージリソースを記述するために使用されます。ストレージ URI はさまざまなノード全体でストレージオブジェクトを一意に識別します。これらは、インターネットで一般的に使用される周知の原則や URI の形式に従います。Oracle Solaris では次のストレージ URI が導入されています。

ローカルデバイス URI

ローカルデバイスストレージ URI タイプは、そのローカルデバイスパスでストレージデバイスを記述します。パスは `/dev` 名前空間内のデバイスを示している必要があります。これらのデバイスは通常、特定のシステムに一意である直接接続ストレージ (DAS) リソースであり、そのデバイス名とパスは一般に移植できません。ただし、これらはまた、複数のノードにわたる統合名前空間を `/dev` の下にすでに用意しているテクノロジーの共有ストレージリソースを指すこともできます。

次に URI 構文のリストを示します。

- `dev:local-path-under-/dev`
- `dev:///path-with-dev`
- `dev:absolute-path-with-dev`

URI 構文の使用例は次のとおりです。

- `dev:dsk/c0t0d0s0`
- `dev:///dev/dsk/c0t0d0`
- `dev:/dev/dsk/c0t0d0`
- `dev:chassis/SYS/HD1/disk`
- `dev:dsk/c0t60A98000564C303132302D6F72613939d0`

ローカルデバイスストレージ URI は、ディスク全体を示す場合も、特定のスライスまたはパーティションを示す場合もあります。ただし、通常、ZFS ストレージプールではスライスまたはパーティションを使用することはお勧めしません。

論理ユニット URI

論理ユニット URI タイプは、ファイバチャネル (FC) またはシリアル接続 SCSI (SAS) ストレージデバイスを記述します。これは、そのデバイス ID (WWN) に基づいた論理ユニット (LU) を示します。論理ユニットストレージ URI は常にディスク全体を表します。

次に URI 構文のリストを示します。

- `lu:luname.naa.ID`
- `lu:luname.eui.ID`
- `lu:initiator.naa.ID,target.naa.ID,luname.naa.ID`
- `lu:initiator.naa.ID,target.naa.ID,luname.eui.ID`

次の例は、URI 構文を使用する方法を示しています。

- `lu:luname.naa.5000c5000288fa25`
- `lu:luname.eui.0021280001cf80f6`
- `lu:initiator.naa.2100001d38089fb0,target.naa.2100001d38089fb0,luname.naa.5000c5000288fa25`
- `lu:initiator.naa.2100001d38089fb0,target.naa.2100001d38089fb0,luname.eui.0021280001cf80f6`

`luname` のみの URI 形式では、ID は論理ユニット名を示します。

`initiator,target,lunname` の形式では、`initiator` は発信元ポートを、`target` はターゲットポートを示します。これらが組み合わさって、論理ユニットへのパスが指定されます。2 番目の構文の論理ユニット名は、最初の構文 (`lunname` のみの形式) の URI の `lunname` と一致する必要があります。

`lunname` のみの URI と組み合わせてマルチパスを使用することをお勧めします。マルチパスが無効で、`lunname` のみの URI を使用する場合は指定した論理ユニットへのランダムなパスが選択されます。パスがランダムに割り当てられないようにするために、`initiator,target,lunname` の URI 形式を使用して論理ユニットへの特定のパスを選択できます。マルチパスが有効で、`initiator,target,lunname` の URI 形式を使用している場合、マルチパスフレームワークによって、論理ユニットにアクセスするために使用するパスが制御され、URI はアクセスパスではなく、ユニットを識別するためにのみ使用されます。

iSCSI URI

iSCSI URI タイプは、iSCSI ネットワークベースのストレージプロトコルを使用してアクセスされるストレージデバイスを記述します。これは常にディスク全体を示します。

iSCSI 修飾名 (IQN) の詳細は、[suri\(5\)](#) のマニュアルページを参照してください。

次に URI 構文のリストを示します。

- `iscsi:///lunname.naa.ID`
- `iscsi:///lunname.eui.ID`
- `iscsi://host[:port]/lunname.naa.ID`
- `iscsi://host[:port]/lunname.eui.ID`
- `iscsi:///target.IQN,lun.LUN`
- `iscsi://host[:port]/target.IQN,lun.LUN`

次の例は、URI 構文を使用する方法を示しています。

- `iscsi:///lunname.eui.0021280001cf80f6`
- `iscsi:///lunname.naa.600144f03d70c80000004ea57da10001`
- `iscsi://[::1]/lunname.naa.600144f03d70c80000004ea57da10001`
- `iscsi://127.0.0.1/lunname.naa.600144f03d70c80000004ea57da10001`
- `iscsi://127.0.0.1:3260/lunname.naa.600144f03d70c80000004ea57da10001`
- `iscsi://hostname:3260/lunname.eui.0021280001cf80f6`
- `iscsi://hostname:3260/lunname.naa.600144f03d70c80000004ea57da10001`
- `iscsi://[::1]/target.iqn.com.sun:02:d0f2d311-f703,lun.2`

- `iscsi://hostname:3620/target.iqn.com.sun:4db41b76-e3d7-cd2f-bf2d-9abef784d76c,lun.0`
- `iscsi:///target.iqn.com.sun:02:d0f2d311-f703,lun.6`
- `iscsi://[::1]/target.iqn.com.sun:02:d0f2d311-f703,lun.2`
- `iscsi://hostname:3620/target.iqn.com.sun:4db41b76-e3d7-cd2f-bf2d-9abef784d76c,lun.0`

luname のみの URI 形式および ID の説明については、[188 ページの「論理ユニット URI」](#) を参照してください。

オプションで、`hostname[:port]` 権限セクションには、認証されていない `SendTargets` 検出アドレスを使用して iSCSI イニシエータを自動的に構成するための情報が記されます。IPv6 アドレスは、角括弧 ([]) で囲む必要があります。

静的または iSNS ベースの構成の場合か、認証を使用する場合にかぎり、luname のみの URI 形式を使用します。これらの関数は、使用する前に、ゾーンフレームワークの外部で構成する必要があります。

iSER トランスポートを介して iSCSI を使用する場合は、iSCSI ストレージの URI は透過的にこれも許可します。iSER を使用するには、ターゲットと発信元は、InfiniBand (IB) パーティションリンクに割り当てられたアドレスを通過する必要があります。大域ゾーンで、`dladm create-part` コマンドを使用して IB パーティションを設定し、`ipadm create-ip` コマンドと `ipadm create-addr` コマンドを使用して IB パーティションにターゲットと発信元のアドレスを割り当てる必要があります。このアドレスは、続いて、ターゲットの検出アドレスを指定するために、iSCSI ストレージ URI の権限セクションで使用されます。

InfiniBand (IB) ハードウェアが存在し、InfiniBand Reliable-Connected (RC) 接続を確立できる場合、iSER 対応の発信元は、iSER 対応のターゲットへの iSER 接続を使用します。RC 接続を確立できない場合、接続は IP ベースの接続を使用して確立されます。

ストレージ URI と共有ストレージリソースの管理

ストレージ URI、またはストレージ URI に基づいた管理者共有ストレージリソースを生成および検証するには、`suriadm` コマンドを使用します。

`suriadm` コマンドを使用して、既存のシステムデバイスパスに基づいて、手動で作成したストレージ URI を検証することも、自動的にストレージ URI を作成することもできます。ストレージ URI タイプに応じて、`suriadm` コマンドを使用して、対応するストレージサブシステムを構成および構成解除できます。ストレージ URI が与えられた場合は、同じコマンドで、ストレージ URI に記述されたストレージオブジェクトの可能なデバイスインスタンスを特定することもできます。次の例は一般的な使用事例に

について説明しています。詳細は、[suriadm\(1M\)](#) のマニュアルページを参照してください。

例 11 suriadm parse 構文を使用したストレージ URI の検証

次の例は、suriadm コマンドを使用してストレージ URI を検証する方法を示します。これらの例では、ストレージ URI の文字列が解析されており、その対応するプロパティが表示されます。検証が終わると、ストレージ URI は後から zonecfg コマンドまたは suriadm コマンドのどちらかで使用できます。

```
root@initiator:~# suriadm parse iscsi://target/lunname.naa.600144F035FF8500000050C884E50001
PROPERTY      VALUE
uri-type      iscsi
hostname      target
port          -
lunname       naa.600144F035FF8500000050C884E50001

root@host:~# suriadm parse dev:/dev/dsk/c4t1d0
PROPERTY      VALUE
uri-type      dev
path          /dev/dsk/c4t1d0
```

例 12 suriadm lookup-uri 構文を使用した、デバイスパスに基づくストレージ URI の生成

次の例は、suriadm コマンドを使用してストレージ URI を作成する方法を示します。この例では、既存のローカルデバイスパスに基づいて、suriadm コマンド出力には、suriadm コマンドか zonecfg コマンドのどちらかであとから使用できる有効なストレージ URI が示されます。

```
root@target:~# suriadm lookup-uri -t iscsi /dev/dsk/c0t600144F035FF8500000050C884E50001d0
iscsi://target/lunname.naa.600144f035ff8500000050c884e50001

root@host:~# suriadm lookup-uri /dev/dsk/c4t1d0
dev:dsk/c4t1d0

root@host:~# suriadm lookup-uri /dev/dsk/c0t600144F0DBF8AF190000510979640005d0
lu:lunname.naa.600144f0dbf8af190000510979640005
lu:initiator.naa.10000000c9991d8c,target.naa.21000024ff3ee89f,lunname.naa.600144f0dbf8af190000510979640005
dev:dsk/c0t600144F0DBF8AF190000510979640005d0
```

例 13 suriadm map 構文を使用した iSCSI ベースのストレージリソースの構成

次の例は、suriadm map コマンドを使用して iSCSI ベースのストレージリソースを構成する方法を示しています。この例では、iSCSI ストレージ URI に対し、suriadm が iSCSI イニシエータの送信先検出アドレスを構成し、iSCSI ターゲットを表すローカルデバイスをインスタンス化します。mapped-dev ストレージ URI プロパティからのローカルデバイスパスは、現在、zpool コマンド、format コマンド、mkfs コマンドなどのユーティリティで使用できます。

```
root@initiator:~# suriadm map iscsi://target/lunname.naa.600144F035FF8500000050C884E50001
PROPERTY      VALUE
```

```
mapped-dev /dev/dsk/c0t600144F035FF850000050C884E50001d0s0
```

例 14 suriadm lookup-mapping 構文を使用した構成済みストレージリソースの検索

この例では、コマンドは、指定されたストレージ URI に現在関連付けられているローカルシステムデバイスを表示するために使用されます。

```
root@initiator:~# suriadm lookup-mapping iscsi://target/luname.naa.600144F035FF850000050C884E50001
PROPERTY      VALUE
mapped-dev    /dev/dsk/c0t600144F035FF850000050C884E50001d0s0
```

例 15 suriadm unmap を使用した iSCSI ベースのストレージリソースの構成解除

suriadm コマンドは、iSCSI ストレージ URI に対して、iSCSI イニシエータの送信先検出アドレスを削除し、共有ストレージリソースを構成解除します。

```
root@initiator:~# suriadm unmap iscsi://target/luname.naa.600144F035FF850000050C884E50001
root@initiator:~# suriadm lookup-mapping iscsi://target/luname.naa.600144F035FF850000050C884E50001
Failed to lookup mapping for URI: "iscsi://target/luname.naa.600144F035FF850000050C884E50001": No such logical
unit name found: "naa.600144F035FF850000050C884E50001"
```

Oracle Solaris ゾーン への共有ストレージリソースの割り当て

ストレージ URI を使用してストレージオブジェクトの場所を記述することにより、ゾーン構成で共有ストレージリソースをゾーンに割り当てます。

2つのゾーン構成リソースタイプ (rootzpool および zpoo1) とプロパティタイプ (storage) は、特定の Oracle Solaris ゾーンに共有ストレージリソースを割り当てるために使用されます。これらは zonecfg で構成および保守されます。

ゾーンの storage プロパティ

共有ストレージリソースを使用するときに、storage プロパティは、ストレージ URI を使用して、ホストに依存しない形式でストレージオブジェクトの場所を定義します。現在、次のストレージ URI が Oracle Solaris の Oracle Solaris ゾーン フレームワークでサポートされています。

- dev: ローカルデバイスパスストレージ URI、DAS
- iscsi: iSCSI ストレージ URI
- lu: ファイバチャネル (FC) およびシリアル接続 SCSI (SAS)

ストレージプロパティは、rootzpool または zpoo1 リソーススコープ内から次の zonecfg サブコマンドを使用して管理されます。

```
zonecfg:zonename:zpool> add storage URI string  
zonecfg:zonename:zpool> remove storage URI string
```

rootzpool リソース

rootzpool リソースは、ゾーンの専用 ZFS ストレージプールです。ゾーンインストール全体が、専用 ZFS ストレージプールにカプセル化されます。この ZFS ストレージプールは共有ストレージリソースから構成されます。

rootzpool リソースは少なくとも 1 つのストレージプロパティを指定する必要があります。複数のストレージプロパティは、冗長な ZFS ストレージプール構成を記述するように指定できます。rootzpool リソースはゾーン構成ごとに 1 つに限られることに注意してください。

rootzpool リソースの ZFS ストレージプール名は、*zonename_rpool* として自動的に割り当てられます。この名前は変更できません。rootzpool リソースは、グローバルリソーススコープから次の zonecfg サブコマンドを使用することによって管理されます。

```
zonecfg:zonename> add rootzpool  
zonecfg:zonename:rootzpool> add storage URI string  
zonecfg:zonename:rootzpool> end  
  
zonecfg:zonename> remove rootzpool  
  
zonecfg:zonename> select rootzpool storage=URI string  
  
zonecfg:zonename> info rootzpool
```

zpool リソース

zpool リソースは、Oracle Solaris ゾーンに委任された共有ストレージリソースから構成される ZFS ストレージプールを記述します。zpool リソースは少なくとも 1 つのストレージプロパティを指定します。複数のストレージプロパティは、冗長な ZFS ストレージプール構成を記述するように指定できます。ゾーン構成に対して複数の zpool リソースを定義できます。

zpool リソースの ZFS ストレージプール名は、ゾーン名と指定の名前プロパティを組み合わせる (*zonename_name* など) ことによって割り当てられます。名前プロパティの場合、zonecfg は、文字列が ZFS ストレージ名と ZFS データセット名として適格であることを検証します。文字列 *rpool* は許可されておらず、このプロパティには使用できません。

zpool リソースは、グローバルリソーススコープから次の zonecfg サブコマンドを使用して管理されます。

```
zonecfg:zonename> add zpool  
zonecfg:zonename:zpool> add storage URI string
```

```
zonecfg:zonename:zpool> set name=name string
zonecfg:zonename:zpool> end

zonecfg:zonename> remove zpool
zonecfg:zonename> remove zpool name=name string
zonecfg:zonename> remove zpool storage=URI string

zonecfg:zonename> select zpool storage=URI string

zonecfg:zonename> info zpool
zonecfg:zonename> info zpool name=name string
zonecfg:zonename> info zpool storage=URI string
```

ゾーンの名前変更

zoneadm コマンドを使用すると、構成済みまたはインストール済みのどちらかの状態のゾーンの名前を変更できます。

ゾーン構成に関する制限事項

既存の共有ストレージリソースの場所の説明が変わったときに rootzpool または zpool リソース内でストレージリソース URI を変更するには、zonecfg remove storage old URI コマンド構文のあとに add storage new URI コマンドを続けて使用します。

共有ストレージリソースでの Oracle Solaris ゾーン の自動 ZFS ストレージプール管理

Oracle Solaris での共有ストレージサポート機能でゾーンを使用する場合、ゾーンフレームワークは、特定のゾーンの rootzpool リソースまたは zpool リソースに関連付けられたすべての ZFS ストレージプールを自動的に管理します。

ゾーンおよびストレージリソースの移行を容易にするため、rootzpool リソースまたは zpool リソースに関連付けられた ZFS ストレージプールは、永続的にはシステムで構成されません。これらは、システムのグローバル ZFS ストレージプールリポジトリ /etc/zfs/zpool キャッシュには表示されません。

ゾーンをインストールまたはクローンすると、ゾーンフレームワークは対応する ZFS ストレージプールを作成します。rootzpool または zpool リソースごとに zonecfg add storage コマンドを使用すると複数の共有ストレージリソースを構成できます。この場合、デフォルトで、ミラー化された ZFS ストレージプールが作成されます。

暗号化などのカスタム構成や、raidz や raidz2 などの異なる冗長性レベルを使用することによって、ZFS ストレージプールをあらかじめ作成できます。最初にゾーン

構成内で必要なすべての共有ストレージリソースを構成したあと、`zpool` ユーティリティを使用して、事前に作成された ZFS ストレージプールを再度エクスポートします。ゾーンフレームワークは最初に、ゾーンのインストールおよびクローニング中に、この事前に作成された ZFS ストレージプールをインポートして使用しようと試みます。

`zoneadm attach` 構文を使用するときに、ゾーンフレームワークはまず共有ストレージリソースをすべて構成し、次に構成済みの ZFS ストレージプールをすべてインポートします。

`zoneadm detach` コマンド構文の場合、ゾーンフレームワークは、まず構成済み ZFS ストレージプールをすべてエクスポートし、次に共有ストレージリソースをすべて構成解除します。

`zoneadm uninstall` コマンドを使用してゾーンをアンインストールする場合、ゾーンフレームワークはデフォルトで、まず構成済み ZFS ストレージプールをすべてエクスポートし、次に共有ストレージリソースをすべて構成解除します。ただし、`zoneadm uninstall` コマンドで `-x force-zpool-destroy` オプションを使用することによって、ZFS ストレージプールを破棄するように明示的に要求できます。

システムブート中に、ゾーン SMF サービス `svc:/system/zones:default` が実行されます。SMF サービスは、共有ストレージリソースを構成し、`rootzpool` リソースまたは `zpool` リソースを含むインストール済み状態のすべてのゾーンに対して ZFS ストレージプールをインポートします。この段階の障害はすべて、対応する SMF サービスログファイル `/var/svc/log/system-zones:default.log` に記録されます。

`rootzpool` リソースまたは `zpool` リソースで構成された ZFS ストレージプールは、ゾーン内部の仮想化された ZFS ストレージプールとして表示されますが、ゾーン管理者が直接管理することはできません。

実行中のゾーンで使用されている間、大域ゾーンで ZFS ストレージプールを破棄またはエクスポートすることはできません。

大域ゾーンから、ゾーンフレームワークで管理される ZFS ストレージプールでの管理アクションに対して `zpool` コマンドを使用できます。管理アクションには、特定のデバイスのオンライン化、オフライン化、障害の発生したデバイスの交換、デバイスの追加または削除などがあります。ゾーン構成を同期し続けるために、使用されている共有ストレージリソースに加えられた変更を反映するように、対応するストレージ URI を適宜更新します。

unavailable 状態について

Oracle Solaris では、`unavailable` のゾーン状態は、ゾーンがインストールされているがブートできないことを示します。この状態は、`zoneadm list -p` コマンド構文の出力に表示されます。

```
root@initiator:~# zoneadm list -cp
0:global:running:/::solaris:shared:-:none
-:iscsi:unavailable:/iscsi:a0a4ba0d-9d6d-cf2c-cc42-f123a5e3ee11:solaris:excl:-:
```

共有ストレージリソースを含むインストール済みゾーンは、ゾーンフレームワークが共有ストレージリソースの構成中の障害や、ZFS ストレージプール管理中の障害に遭遇した場合、使用不可の状態に入ることがあります。

使用不可の状態からゾーンを移すには、最初に共有ストレージリソースの接続やゾーンの構成ミスに関連する問題を特定し、おそらく修正する必要があります。次に `zoneadm attach` コマンドを使用して、ゾーンを適切に再接続し、インストール済みのゾーン状態に移すことができます。 `zoneadm uninstall` コマンドで、ゾーンをアンインストールし、構成済みのゾーン状態に戻すことも可能です。

ゾーン状態の詳細は、『[Oracle Solaris ゾーン](#)の紹介』の「[非大域ゾーンの状態モデル](#)」を参照してください。

追加の zoneadm サブコマンドオプション

該当する場合は、zoneadm CLI サブコマンドは、ZFS ストレージプール管理専用のオプションで拡張されています。

ゾーンのインストール、クローニング、接続用のオプション

以降のセクションでは、zoneadm の `install`、`clone`、および `attach` サブコマンドの3つの使用可能なオプションについて説明します。

-x force-zpool-import オプション

このオプションは、ゾーン構成で指定されたすべての `zpool` リソースに適用され、(たとえば、別のシステムによって) 使用されているように見える可能性のあるすべての ZFS ストレージプールを強制的にインポートするようゾーンフレームワークに指示します。これは、`zpool import -f` コマンドの動作を模倣します。

このオプションは、たとえば、`zoneadm detach` コマンドや `zoneadm uninstall` コマンドを使用して ZFS ストレージプールが別のシステムに適切にエクスポートされなかった場合に、ゾーンの移行中に新しいシステムに ZFS ストレージプールを簡単にインポートできるようにします。これらの状況において、特定のゾーンに対して構成されたすべての ZFS ストレージプールに、強制インポートを適用する必要があります。

-x force-zpool-create= オプション

このオプションは、既存の古いプールに対して、またはほかの目的で現在使用されているように見えるデバイス上に新しい ZFS ストレージプールを強制的に作成するように、ゾーンフレームワークに対して指示します。これは、`zpool create -f` コマンド構文の動作を模倣します。

このオプションの範囲は、たとえば、`zpool` 名のコンマ区切りのリストで指定されたり、オプションを複数回使用して指定された ZFS ストレージプールに限られます。

```
-x force-zpool-create=rpool,pool2,pool3
-x force-zpool-create=pool1 -x force-zpool-create=pool2
```

ここで使用される ZFS ストレージプール名は、ゾーン構成内の対応する `zpool` リソースの名前プロパティです。`rootzpool` リソースによって記述された ZFS ストレージプールを指定するには、`rpool` の名前を使用します。

-x force-zpool-create-all オプション

このオプションは、ゾーン構成で指定されたすべての `zpool` リソースおよび `rootzpool` リソースに対して新しい ZFS ストレージプールを強制的に作成するように、ゾーンフレームワークに指示します。

ゾーンのアンインストール用オプション

`zoneadm uninstall` サブコマンドでは、次の 3 つのオプションを使用できます。

-x force-zpool-destroy=zpoolname オプション

このオプションは、ゾーンをアンインストールするときに、ZFS ストレージプールを破棄するように、ゾーンフレームワークに指示します。これは、`zpool destroy poolname` コマンドの動作を模倣します。このオプションには `zpoolname` リストを指定します。

```
force-zpool-destroy=zpoolname{,zpoolname,zpoolname,...}
```

このオプションの範囲は、`zpool` 名のコンマ区切りリストで指定されるか、またはこのオプションを複数回使用して指定される ZFS ストレージプールに制限されません。次に例を示します。

```
-x force-zpool-destroy=rpool,pool2,pool3
-x force-zpool-destroy=pool1 -x force-zpool-destroy=pool2
```

名前引数は、目的の `zpool` リソースについてゾーン構成で指定された名前プロパティに対応します。名前 `rpool` は、`rootzpool` リソースに関連付けられた ZFS ストレージプールを指定するために使用されます。

-x force-zpool-destroy-all オプション

このオプションは、ゾーンフレームワークに、ゾーンのアンインストール時にゾーン構成で指定されているすべての `zpool` および `rootzpool` リソースの ZFS ストレージプールを破棄するよう指示します。

ゾーンのアンインストール時に ZFS ストレージプールを破棄するようにゾーンフレームワークに強制するオプションは、慎重に使用する必要があります。現在アクティブなブート環境 (BE) でゾーンをアンインストールする場合でも、現在非アクティブなブート環境に所属しているほかのゾーンブート環境 (ZBE) が、この ZFS ストレージプール内に存在する可能性があります。`rootzpool` リソースに関連付けられた ZFS ストレージプールを破棄すると、インストール済み状態でゾーンを保持するすべてのブート環境で、ゾーンが利用できなくなります。ゾーンブート環境の詳細は、[beadm\(1M\)](#) のマニュアルページを参照してください。

-x force-storage-destroy-all オプション

ストレージを破棄するには、このオプションを指定します。すべてのストレージ URI タイプがこの操作をサポートしているわけではないことに注意してください。ファイルまたは NFS ストレージ URI によって参照されているストレージを破棄できますが、`dev`、`iSCSI`、または `lu` ストレージ URI によって参照されているストレージは破棄できません。

zoneadm コマンドの使用に関する制限事項

`rootzpool` リソースで構成されたゾーンの場合、`zoneadm move` コマンド構文は、`zonpath` の名前変更の許可だけに制限されています。`rootzpool` リソースを含むゾーンを、対応する ZFS ストレージプールからファイルシステム階層の別の場所または別の ZFS ストレージプールに移動させることはできません。

共有ストレージリソースでホストされる Oracle Solaris ゾーンの実装

次のセクションでは、共有ストレージリソースでの Oracle Solaris ゾーンの配備について説明します。

共有ストレージ上のゾーンに関する zpoo1 構成の考慮事項

ZFS ストレージプールを構成する場合、ミラー、RAIDZ、RAIDZ-2 などの冗長なプール構成を使用することができます。共有ストレージリソースをホストするバックエンドストレージで、たとえばハードウェア RAID の形式で障害許容力およびデータ保護が確保されている場合でも、冗長な構成によって、ZFS ストレージプールは固有の自己回復機能を使用できるようになります。

Oracle Solaris ゾーン を共有ストレージリソースでホストする冗長な ZFS ストレージプール構成を使用する 2 つの方法があります。

- ミラー化した ZFS ストレージプール構成を使用できます。これはデフォルトで、`rootzpool` リソースまたは `zpool` リソースのストレージプロパティをゾーン構成で複数指定した場合、自動的に作成されます。
- または、`zpoo1 create` コマンドで、RAIDZ や RAIDZ-2 などのより複雑な ZFS ストレージプール構成をあらかじめ作成しておくこともできます。`zpoo1 export` を使用して、新しく作成した ZFS ストレージプールを再度エクスポートする必要がありますことに注意してください。次に、`zonecfg` コマンドを使用して、ゾーン構成で、すべての対応するストレージリソース URI を `rootzpool` リソースまたは `zpool` リソースに追加する必要があります。ゾーンフレームワークは次に、新しい ZFS ストレージプールを作成しようと試みるのではなく、ゾーンのインストールまたはクローン中にこの ZFS ストレージプールをインポートします。

サンプルシナリオ

このセクションでは、共有ストレージリソースに Oracle Solaris ゾーン を配備する例を示します。

例 16 iSCSI ベースの共有ストレージデバイスを使用した Oracle Solaris ゾーン

この演習では、iSCSI ターゲット経由で共有ストレージを提供するサンプル構成を Oracle Solaris サーバー上で設定します。ゾーンは Oracle Solaris が動作している 2 番目のサーバーに構成およびインストールされ、iSCSI ベースの共有ストレージリソースを使用してゾーンをホストします。

最初に、次の `pkg install` コマンドのいずれかを使用して、対応するパッケージをインストールします。最初のコマンドは、マルチプロトコルストレージサーバーグループパッケージ全体をインストールします。`itadm(1M)` および `stmfadm(1M)` のマニュアルページに説明されているように、2 番目のコマンドは、一般的なマルチプロトコル SCSI ターゲット (COMSTAR) フレームワーク内で iSCSI のターゲットサポートだけをインストールします。

```
root@target:~# pkg install group/feature/storage-server
root@target:~# pkg install system/storage/iscsi/iscsi-target
```

次に、このサーバーから iSCSI ターゲットをエクスポートするためのバッキングストアを作成します。zfs コマンドを使用して、4つの iSCSI ターゲット論理ユニット用のバッキングストアとして、ターゲットサーバーの rpool/export データセット内に格納される4つの ZFS ボリューム (サイズはそれぞれ 10G バイト) を作成します。

```
root@target:~# zfs create -V 10G rpool/export/zonevol1
root@target:~# zfs create -V 10G rpool/export/zonevol2
root@target:~# zfs create -V 10G rpool/export/zonevol3
```

バッキングストアを設定したあと、stmfadm コマンドを使用して、各 ZFS ボリュームのターゲット論理ユニットを作成します。これにより、それぞれに対応したデバイス ID (WWN) が与えられ、後からストレージ URI でクライアントホストでの iSCSI ターゲット検出に使用されます。

```
root@target:~# stmfadm create-lu /dev/zvol/rdisk/rpool/export/zonevol1
Logical unit created: 600144F035FF8500000050C884E50001
root@target:~# stmfadm create-lu /dev/zvol/rdisk/rpool/export/zonevol2
Logical unit created: 600144F035FF8500000050C884E80002
root@target:~# stmfadm create-lu /dev/zvol/rdisk/rpool/export/zonevol3
Logical unit created: 600144F035FF8500000050C884EC0003
```

stmfadm list-lu 構文で、構成された論理ユニットを表示できます。

```
root@target:~# stmfadm list-lu
LU Name: 600144F035FF8500000050C884E50001
LU Name: 600144F035FF8500000050C884E80002
LU Name: 600144F035FF8500000050C884EC0003
LU Name: 600144F035FF8500000050C884EC0004
```

stmfadm list-lu -v 構文で、構成済みの論理ユニットの詳細を照会できます。

```
root@target:~# stmfadm list-lu -v
LU Name: 600144F035FF8500000050C884E50001
Operational Status      : Online
Provider Name           : sbd
Alias                    : /dev/zvol/rdisk/rpool/export/zonevol1
View Entry Count        : 0
Data File                : /dev/zvol/rdisk/rpool/export/zonevol1
Meta File                : not set
Size                     : 10737418240
Block Size               : 512
Management URL          : not set
Software ID              : not set
Vendor ID                : SUN
Product ID               : COMSTAR
Serial Num               : not set
Write Protect            : Disabled
Write Cache Mode Select : Enabled
Writeback Cache          : Enabled
Access State             : Active
```

論理ユニットを iSCSI イニシエータで使用できるようにするには、stmfadm add-view コマンドで、ターゲットサーバーに論理ユニットビューを追加します。

```
root@target:~# stmfadm add-view 600144F035FF8500000050C884E50001
root@target:~# stmfadm add-view 600144F035FF8500000050C884E80002
root@target:~# stmfadm add-view 600144F035FF8500000050C884EC0003
```

次に、ターゲットサーバー上で iSCSI ターゲットを構成します。最初に `svcadm enable` を使用して、iSCSI ターゲット SMF サービスを有効にします。

```
root@target:~# svcadm enable -r svc:/network/iscsi/target:default
```

次に、`itadm create-target` を使用して iSCSI ターゲット自体を作成します。

```
root@target:~# itadm create-target
Target iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a successfully created
```

`itadm list-target` または `stmfadm list-target` のどちらかを使用すると、構成済みの iSCSI ターゲットに関する詳細を照会できます。

```
root@target:~# itadm list-target -v
TARGET NAME                                STATE    SESSIONS
iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a  online  0
  alias:                                     -
  auth:                                       none (defaults)
  targetchapuser:                            -
  targetchapsecret:                          unset
  tpg-tags:                                   default
```

```
root@target:~# stmfadm list-target -v
Target: iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a
Operational Status      : Online
Provider Name           : iscsit
Alias                    : -
Protocol                 : iSCSI
Sessions                 : 0
```

2 番目のサーバー上のゾーン構成で使用される対応するストレージ URI を取得するには、`suriadm(1M)` を使用します。論理ユニットごとに、ローカルデバイスパスエントリが `/dev` に作成されています。`suriadm` コマンドは、iSCSI ストレージ URI の作成に使用されます。

```
root@target:~# suriadm lookup-uri -t iscsi /dev/dsk/c0t600144F035FF8500000050c884E50001d0
iscsi://target/lunname.naa.600144f035ff8500000050c884e50001
```

```
root@target:~# suriadm lookup-uri -t iscsi /dev/dsk/c0t600144F035FF8500000050c884E80002d0
iscsi://target/lunname.naa.600144f035ff8500000050c884e80002
```

```
root@target:~# suriadm lookup-uri -t iscsi /dev/dsk/c0t600144F035FF8500000050c884EC0003d0
iscsi://target/lunname.naa.600144f035ff8500000050c884ec0003
```

`stmfadm(1M)` に記載されている `stmfadm` コマンドからの GUID、および `itadm(1M)` に記載されている `stmfadm` コマンドからの iSCSI ターゲット IQN に基づいた iSCSI URI を取得するには、次のようにします。

```
root@accelerated:~# suriadm lookup-uri -t iscsi -p
target=iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a -p lun=0
iscsi://accelerated/lunname.naa.600144f035ff8500000050c884ef0004
iscsi://accelerated/target.iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a,lun.0
```

```
root@accelerated:~# suriadm lookup-uri -t iscsi -p
target=iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a -p lun=1
iscsi://accelerated/lunname.naa.600144f035ff8500000050c884ec0003
iscsi://accelerated/target.iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a,lun.1
```

```
root@accelerated:~# suriadm lookup-uri -t iscsi -p
```

```
target=iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a -p lun=2
iscsi://accelerated/lunname.naa.600144f035ff8500000050c884e80002
iscsi://accelerated/target.iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e938
36ad0d6a,lun.2

root@accelerated:~# suriadm lookup-uri -t iscsi -p
target=iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e93836ad0d6a -p lun=3
iscsi://accelerated/lunname.naa.600144f035ff8500000050c884e50001
iscsi://accelerated/target.iqn.1986-03.com.sun:02:b62a8291-b89e-41ba-9aef-e938
36ad0d6a,lun.3
```

これにより、iSCSI ターゲットストレージを提供するサンプルサーバー上で実行されるタスクが完了します。

次の手順は、iSCSI 経由で提供されるこの共有ストレージを使用して、2 番目のサーバー上でゾーンを構成およびインストールするためのものです。

最初の手順として、iSCSI イニシエータとして選択されたクライアントサーバー上で対応するパッケージをインストールします。

```
root@initiator:~# pkg install pkg:/system/storage/iscsi/iscsi-initiator
```

次に、zonecfg コマンドを使用して、rootzpool リソースおよび zpoo1 リソースでゾーンを構成します。共有ストレージリソースとして構成した 3 つの iSCSI ターゲット論理ユニットを使用して、そのゾーンをホストします。ターゲットサーバー上で、以前に suriadm で取得した iSCSI ストレージ URI を使用します。

```
root@initiator:~# zonecfg -z iscsi
Use 'create' to begin configuring a new zone.
zonecfg:iscsi> create
create: Using system default template 'SYSdefault'
zonecfg:iscsi> set zonepath=/iscsi
zonecfg:iscsi> add rootzpool
zonecfg:iscsi:rootzpool> add storage iscsi://target/lunname.naa.600144F035FF8500000050C884E50001
zonecfg:iscsi:rootzpool> end
zonecfg:iscsi> add zpool
zonecfg:iscsi:zpool> set name=data
zonecfg:iscsi:zpool> add storage iscsi://target/lunname.naa.600144F035FF8500000050C884E80002
zonecfg:iscsi:zpool> add storage iscsi://target/lunname.naa.600144F035FF8500000050C884EC0003
zonecfg:iscsi:zpool> end
zonecfg:iscsi> commit
zonecfg:iscsi> exit
```

zoneadm install を使用してゾーンをインストールします。

```
root@initiator:~# zoneadm -z iscsi install
Configured zone storage resource(s) from:
  iscsi://target/lunname.naa.600144F035FF8500000050C884E50001
Created zone zpool: iscsi_rpool
Configured zone storage resource(s) from:
  iscsi://target/lunname.naa.600144F035FF8500000050C884E80002
  iscsi://target/lunname.naa.600144F035FF8500000050C884EC0003
Created zone zpool: iscsi_data
Progress being logged to /var/log/zones/zoneadm.20130125T112209Z.iscsi.install
Image: Preparing at /iscsi/root.

AI Manifest: /tmp/manifest.xml.pmai7h
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Zonename: iscsi
Installation: Starting ...
```

```

Creating IPS image
Startup linked: 1/1 done
Installing packages from:
solaris
origin: http://pkg.oracle.com/solaris/release/
DOWNLOAD          PKGS      FILES    XFER (MB)   SPEED
Completed          183/183   33556/33556 222.2/222.2 3.4M/s

PHASE              ITEMS
Installing new actions 46825/46825
Updating package state database      Done
Updating image state                 Done
Creating fast lookup database        Done
Installation: Succeeded

```

Note: Man pages can be obtained by installing pkg:/system/manual done.

Done: Installation completed in 266.487 seconds.

Next Steps: Boot the zone, then log into the zone console (zlogin -C)

to complete the configuration process.

```

Log saved in non-global zone as /iscsi/root/var/log/zones/
zoneadm.20130125T112209Z.iscsi.install
root@initiator:~#

```

ゾーンのインストールが完了したら、`zoneadm list` を使用して、ゾーンが正しくインストールされたことを確認します。

```

root@initiator:~# zoneadm list -cp
0:global:running:/::solaris:shared:-:none
-:iscsi:installed:/iscsi:a0a4ba0d-9d6d-cf2c-cc42-f123a5e3ee11:solaris:excl:-:

```

最後に、`zpool` コマンドを使用して、このゾーンに関連付けられた、新しく作成した ZFS ストレージプールを監視します。

```

root@initiator:~# zpool list
NAME          SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
iscsi_data    9.94G  83.5K  9.94G  0%  1.00x  ONLINE  -
iscsi_rpool   9.94G   436M  9.51G  4%  1.00x  ONLINE  -

```

```

root@initiator:~# zpool status -v iscsi_rpool
pool: iscsi_rpool
state: ONLINE
scan: none requested
config:

```

NAME	STATE	READ	WRITE	CKSUM
iscsi_rpool	ONLINE	0	0	0
c0t600144F035FF850000050C884E50001d0	ONLINE	0	0	0

```

root@initiator:~# zpool status -v iscsi_data
pool: iscsi_data
state: ONLINE
scan: none requested
config:

```

NAME	STATE	READ	WRITE	CKSUM
iscsi_data	ONLINE	0	0	0

```

mirror-0                ONLINE      0      0      0
  c0t600144F035FF850000050C884E80002d0 ONLINE      0      0      0
  c0t600144F035FF850000050C884EC0003d0 ONLINE      0      0      0

```

ゾーンのインストール全体がこの ZFS ストレージプール内に含まれます。このゾーンの ZFS データセットレイアウトは次のとおりです。

```

root@initiator:~# zfs list -t all|grep iscsi
iscsi_data                83.5K  9.78G   31K  /iscsi_data
iscsi_rpool               436M  9.36G   32K  /iscsi
iscsi_rpool/rpool        436M  9.36G   31K  /rpool
iscsi_rpool/rpool/ROOT   436M  9.36G   31K  legacy
iscsi_rpool/rpool/ROOT/solaris 436M  9.36G  390M  /iscsi/root
iscsi_rpool/rpool/ROOT/solaris@install 64K   -    390M  -
iscsi_rpool/rpool/ROOT/solaris/var 46.1M  9.36G  45.4M  /iscsi/root/var
iscsi_rpool/rpool/ROOT/solaris/var@install 644K   -    45.4M  -
iscsi_rpool/rpool/VARSHARE 31K   9.36G   31K  /var/share
iscsi_rpool/rpool/export 62K   9.36G   31K  /export
iscsi_rpool/rpool/export/home 31K   9.36G   31K  /export/home

```

iSCSI ベースの共有ストレージリソースでホストされる新しいゾーンは正しくインストールされており、`zoneadm(1M) boot` を使用してブートすることができるようになっています。

ゾーンがブートされたあと、ゾーン管理者は、仮想化された ZFS データセットおよびストレージプールをゾーン内から監視します。

```

root@iscsi:~# zpool list
NAME    SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
data    9.94G  85K    9.94G  0%   1.00x  ONLINE  -
rpool   9.94G  449M   9.50G  4%   1.00x  ONLINE  -

root@iscsi:~# zpool status -v
pool: data
state: ONLINE
scan: none requested
config:

        NAME                                STATE      READ WRITE CKSUM
        data                                ONLINE    0     0   0
        mirror-0                            ONLINE    0     0   0
        c0t600144F035FF850000050C884E80002d0 ONLINE    0     0   0
        c0t600144F035FF850000050C884EC0003d0 ONLINE    0     0   0

pool: rpool
state: ONLINE
scan: none requested
config:

        NAME                                STATE      READ WRITE CKSUM
        rpool                                ONLINE    0     0   0
        c0t600144F035FF850000050C884E50001d0 ONLINE    0     0   0

root@iscsi:~# zfs list -t all
NAME                                USED  AVAIL  REFER  MOUNTPOINT
data                                85K   9.78G   31K    /data
rpool                               464M   9.33G   31K    /rpool
rpool/ROOT                          464M   9.33G   31K    legacy
rpool/ROOT/solaris                  464M   9.33G  416M    /
rpool/ROOT/solaris@install          1.83M   -    390M    -
rpool/ROOT/solaris/var              46.2M   9.33G  45.6M  /var
rpool/ROOT/solaris/var@install      674K   -    45.4M  -

```

```

rpool/VARSHARE          39K  9.33G   39K  /var/share
rpool/export            96.5K 9.33G   32K  /export
rpool/export/home      64.5K 9.33G   32K  /export/home
rpool/export/home/user 32.5K 9.33G  32.5K /export/home/user

```

例 17 DAS ストレージデバイスを使用した Oracle Solaris ゾーン の例

この演習では、直接接続したローカルストレージデバイスを使用して、Oracle Solaris 上でゾーンを構成しインストールします。この方法は通常、異なるホスト間で移植できないことに注意してください。

最初に、`format` コマンドを使用して、利用可能なローカルディスクを検出します。次に、`suriadm lookup-uri` を使用して、ゾーン構成内で使用される対応するストレージ URI を構築します。

```

root@host:~# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
   1. c4t1d0 <SEAGATE-ST336704LSUN36G-0326-33.92GB>
      /pci@0,0/pci1022,7450@a/pci17c2,20@4/sd@1,0
   2. c4t2d0 <FUJITSU-MAT3073NC-0104-68.49GB>
      /pci@0,0/pci1022,7450@a/pci17c2,20@4/sd@2,0
   3. c4t3d0 <SEAGATE-ST336704LSUN36G-0326-33.92GB>
      /pci@0,0/pci1022,7450@a/pci17c2,20@4/sd@3,0
   4. c4t4d0 <FUJITSU-MAW3073NC-0103-68.49GB>
      /pci@0,0/pci1022,7450@a/pci17c2,20@4/sd@4,0

root@host:~# suriadm lookup-uri -t dev /dev/dsk/c4t1d0
dev:dsk/c4t1d0
root@host:~# suriadm lookup-uri -t dev /dev/dsk/c4t2d0
dev:dsk/c4t2d0
root@host:~# suriadm lookup-uri -t dev /dev/dsk/c4t3d0
dev:dsk/c4t3d0
root@host:~# suriadm lookup-uri -t dev /dev/dsk/c4t4d0
dev:dsk/c4t4d0

```

これらのストレージ URI を使用して、`rootzpool` および `zpool` リソース (どちらもミラー化された ZFS ストレージプールを示す) でゾーンを構成します。

```

root@host:~# zonecfg -z disk
Use 'create' to begin configuring a new zone.
zonecfg:disk> create
create: Using system default template 'SYSdefault'
zonecfg:disk> set zonepath=/disk
zonecfg:disk> add rootzpool
zonecfg:disk:rootzpool> add storage dev:dsk/c4t1d0
zonecfg:disk:rootzpool> add storage dev:dsk/c4t3d0
zonecfg:disk:rootzpool> end
zonecfg:disk> add zpool
zonecfg:disk:zpool> set name=dpool
zonecfg:disk:zpool> add storage dev:dsk/c4t2d0
zonecfg:disk:zpool> add storage dev:dsk/c4t4d0
zonecfg:disk:zpool> end
zonecfg:disk> commit
zonecfg:disk> exit

```

次にゾーンをインストールします。

```

root@host:~# zoneadm -z disk install

```

```

Created zone zpool: disk_rpool
Created zone zpool: disk_dpool
Progress being logged to /var/log/zones/zoneadm.20130213T132236Z.disk.install
Image: Preparing at /disk/root.

AI Manifest: /tmp/manifest.xml.r0a0he
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Zonename: disk
Installation: Starting ...

Creating IPS image
Startup linked: 1/1 done
Installing packages from:
solaris
origin: http://pkg.oracle.com/solaris/release/
DOWNLOAD          PKGS      FILES    XFER (MB)   SPEED
Completed          183/183   33556/33556 222.2/222.2 2.0M/s

PHASE              ITEMS
Installing new actions      46825/46825
Updating package state database      Done
Updating image state          Done
Creating fast lookup database        Done
Installation: Succeeded

Note: Man pages can be obtained by installing pkg:/system/manual
done.

Done: Installation completed in 308.358 seconds.

Next Steps: Boot the zone, then log into the zone console (zlogin -C)
to complete the configuration process.

Log saved in non-global zone as /disk/root/var/log/zones/
zoneadm.20130213T132236Z.disk.install
root@host:~#

```

ゾーンのインストール後、次の2つの新しいZFSストレージプールがオンラインになります。

```

root@host:~# zpool list
NAME          SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
disk_dpool    68G   83.5K  68.0G   0%  1.00x  ONLINE  -
disk_rpool    33.8G  434M   33.3G   1%  1.00x  ONLINE  -

root@host:~# zpool status -v disk_rpool
pool: disk_rpool
state: ONLINE
scan: none requested
config:

NAME          STATE      READ WRITE CKSUM
disk_rpool    ONLINE    0     0     0
mirror-0     ONLINE    0     0     0
  c4t1d0     ONLINE    0     0     0
  c4t3d0     ONLINE    0     0     0

root@host:~# zpool status -v disk_dpool
pool: disk_dpool
state: ONLINE
scan: none requested
config:

```

```

NAME          STATE      READ WRITE CKSUM
disk_dpools   ONLINE    0    0    0
mirror-0     ONLINE    0    0    0
  c4t2d0     ONLINE    0    0    0
  c4t4d0     ONLINE    0    0    0

```

ゾーンのインストール全体がこの ZFS ストレージプール内に含まれます。このゾーンの ZFS データセットレイアウトは次のとおりです。

```

root@host:~# zfs list -t all|grep disk
disk_dpools           83.5K 66.9G   31K /disk_dpools
disk_rpool            434M 32.8G   32K /disk
disk_rpool/rpool     433M 32.8G   31K /rpool
disk_rpool/rpool/ROOT 433M 32.8G   31K legacy
disk_rpool/rpool/ROOT/solaris 433M 32.8G 389M /disk/root
disk_rpool/rpool/ROOT/solaris@install 63K - 389M -
disk_rpool/rpool/ROOT/solaris/var 43.8M 32.8G 43.2M /disk/root/var
disk_rpool/rpool/ROOT/solaris/var@install 584K - 43.2M -
disk_rpool/rpool/VARSHARE 31K 32.8G 31K /var/share
disk_rpool/rpool/export 62K 32.8G 31K /export
disk_rpool/rpool/export/home 31K 32.8G 31K /export/home

```

ローカルデバイスストレージリソースでホストされる新しいゾーンは正しくインストールされており、`zoneadm boot` コマンドを使用してブートできるようになっています。

ゾーンがブートされたあと、ゾーン管理者は、仮想化された ZFS データセットおよびストレージプールをゾーン内部から監視できます。

```

root@disk:~# zpool list
NAME  SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
dpool 68G   83.5K 68.0G 0%   1.00x  ONLINE  -
rpool 33.8G 472M   33.3G 1%   1.00x  ONLINE  -

```

```

root@disk:~# zpool status -v
pool: dpool
state: ONLINE
scan: none requested
config:

```

```

NAME          STATE      READ WRITE CKSUM
dpool         ONLINE    0    0    0
  mirror-0   ONLINE    0    0    0
    c4t2d0   ONLINE    0    0    0
    c4t4d0   ONLINE    0    0    0

```

```

pool: rpool
state: ONLINE
scan: none requested
config:

```

```

NAME          STATE      READ WRITE CKSUM
rpool         ONLINE    0    0    0
  mirror-0   ONLINE    0    0    0
    c4t1d0   ONLINE    0    0    0
    c4t3d0   ONLINE    0    0    0

```

```

root@disk:~# zfs list -t all
NAME          USED  AVAIL  REFER  MOUNTPOINT
dpool         83.5K 66.9G   31K    /dpool
rpool         465M 32.8G   31K    /rpool

```

```

rpool/ROOT                465M 32.8G 31K legacy
rpool/ROOT/solaris        465M 32.8G 416M /
rpool/ROOT/solaris@install 5.60M - 389M -
rpool/ROOT/solaris/var    43.9M 32.8G 43.3M /var
rpool/ROOT/solaris/var@install 618K - 43.2M -
rpool/VARSHARE            39K 32.8G 39K /var/share
rpool/export              96.5K 32.8G 32K /export
rpool/export/home        64.5K 32.8G 32K /export/home
rpool/export/home/user   32.5K 32.8G 32.5K /export/home/user

```

例 18 ファイバチャネルベースのストレージデバイスを使用した Oracle Solaris ゾーン

この演習では、ファイバチャネルを介して提供される共有ストレージデバイスを使用して、Oracle Solaris 上でゾーンを構成しインストールします。

最初に、`fcinfo lu` コマンドを使用することによって、現在ホストから確認できるファイバチャネルの論理ユニットを検出します。

```

root@host:~# fcinfo lu -v
OS Device Name: /dev/rdisk/c0t600144F0DBF8AF190000510979640005d0s2
      HBA Port WWN: 10000000c9991d8c
      Remote Port WWN: 21000024ff3ee89f
      LUN: 5
Vendor: SUN
Product: ZFS Storage 7120
Device Type: Disk Device

```

`suriadm lookup-uri` を使用して、デバイスパスに基づいてストレージ URI を構築します。クエリーで LU 全体を表すストレージ URI を取得できるように、デバイス名のスライス部分を削除します。

```

root@host:~# suriadm lookup-uri /dev/dsk/c0t600144F0DBF8AF190000510979640005d0
lu:luname.naa.600144f0dbf8af190000510979640005
lu:initiator.naa.10000000c9991d8c,target.naa.21000024ff3ee89f,luname.naa.
600144f0dbf8af190000510979640005
dev:dsk/c0t600144F0DBF8AF190000510979640005d0

```

表示された 3 つの URI から、ゾーン構成で使用する、`luname` のみの形式の論理ユニットストレージ URI を選択します。

```

root@host:~# zonecfg -z fc
Use 'create' to begin configuring a new zone.
zonecfg:fc> create
create: Using system default template 'SYSdefault'
zonecfg:fc> set zonepath=/fc
zonecfg:fc> add rootzpool
zonecfg:fc:rootzpool> add storage lu:luname.naa.600144f0dbf8af190000510979640005
zonecfg:fc:rootzpool> end
zonecfg:fc> commit
zonecfg:fc> exit

```

ゾーンをインストールします。

```

root@host:~# zoneadm -z fc install
Created zone zpool: fc_rpool
Progress being logged to /var/log/zones/zoneadm.20130214T045957Z.fc.install
Image: Preparing at /fc/root.

```

```

AI Manifest: /tmp/manifest.xml.K9aaow
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
  Zonename: fc
Installation: Starting ...

      Creating IPS image
Startup linked: 1/1 done
      Installing packages from:
        solaris
          origin: http://pkg.oracle.com/solaris/release/
DOWNLOAD          PKGS          FILES          XFER (MB)   SPEED
Completed          190/190      34246/34246    231.3/231.3  7.2M/s

PHASE              ITEMS
Installing new actions      48231/48231
Updating package state database      Done
Updating image state          Done
Creating fast lookup database        Done
Installation: Succeeded

      Note: Man pages can be obtained by installing pkg:/system/manual
done.

      Done: Installation completed in 104.318 seconds.

      Next Steps: Boot the zone, then log into the zone console (zlogin -C)
                  to complete the configuration process.

Log saved in non-global zone as /fc/root/var/log/zones/zoneadm.20130214T045957Z.fc.install
root@host:~#

```

ゾーンのインストール後、次の新しい ZFS ストレージプールがオンラインになります。

```

root@host:~# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
fc_rpool 39.8G  441M   39.3G  1%  1.00x  ONLINE  -

root@host:~# zpool status -v fc_rpool
pool: fc_rpool
state: ONLINE
scan: none requested
config:

NAME                STATE          READ WRITE CKSUM
fc_rpool             ONLINE         0     0     0
  c0t600144F0DBF8AF190000510979640005d0  ONLINE         0     0     0

```

ゾーンのインストール全体がこの ZFS ストレージプール内に含まれます。ゾーンには次の ZFS データセットレイアウトがあります。

```

root@host:~# zfs list -t all|grep fc
fc_rpool             440M  38.7G   32K  /fc
fc_rpool/rpool      440M  38.7G   31K  /rpool
fc_rpool/rpool/ROOT 440M  38.7G   31K  legacy
fc_rpool/rpool/ROOT/solaris 440M  38.7G  405M  /fc/root
fc_rpool/rpool/ROOT/solaris@install 67K   -    405M  -
fc_rpool/rpool/ROOT/solaris/var 34.3M 38.7G  33.6M  /fc/root/var
fc_rpool/rpool/ROOT/solaris/var@install 665K  -    33.6M  -
fc_rpool/rpool/VARSHARE 31K   38.7G   31K  /var/share
fc_rpool/rpool/export 62K   38.7G   31K  /export

```

```
fc_rpool/rpool/export/home          31K 38.7G 31K /export/home
```

ファイバチャネルターゲットから提供される共有ストレージでホストされる新しいゾーンは、正しくインストールされました。これで、`zoneadm boot` を使用してこのゾーンをブートできます。

ゾーンがブートされたあと、ゾーン管理者は、仮想化された ZFS データセットおよびストレージプールをゾーン内部から監視できます。

```
root@fc:~# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTRoot
rpool    39.8G  451M   39.3G   1%  1.00x  ONLINE  -

root@fc:~# zpool status -v
pool: rpool
state: ONLINE
scan: none requested
config:

          NAME                                 STATE      READ WRITE CKSUM
rpool    ONLINE
c0t600144F0DBF8AF190000510979640005d0  ONLINE      0     0     0

root@fc:~# zfs list -t all
NAME                                     USED  AVAIL  REFER  MOUNTPOINT
rpool                                   467M  38.7G   31K    /rpool
rpool/ROOT                             467M  38.7G   31K    legacy
rpool/ROOT/solaris                     467M  38.7G  430M    /
rpool/ROOT/solaris@install             1.90M  -      405M    -
rpool/ROOT/solaris/var                 34.4M  38.7G  33.7M    /var
rpool/ROOT/solaris/var@install         703K  -      33.6M    -
rpool/VARSHARE                          39K   38.7G   39K    /var/share
rpool/export                           96.5K  38.7G   32K    /export
rpool/export/home                       64.5K  38.7G   32K    /export/home
rpool/export/home/user                   32.5K  38.7G   32.5K   /export/home/user
```

共有ストレージリソースでホストされる Oracle Solaris ゾーン移行

ゾーン構成のストレージ URI を使用した共有ストレージでホストされるゾーンの移行は、明瞭で単純なプロセスです。このプロセスに必要な唯一の CLI は、`zoneadm` コマンドと `zonecfg` コマンドです。ストレージリソースまたは ZFS ストレージプールを含むゾーンの移行には、追加コマンドを実行する必要はありません。

例 19 iSCSI 共有ストレージに基づくゾーンの移行

この例では、iSCSI 共有ストレージに基づいて、現在のホストから新しいホストにゾーンを移行するための手順を示します。

```
root@initiator:~# zoneadm list -cp
0:global:running:/::solaris:shared:::none
-:iscsi:installed:/iscsi:a0a4ba0d-9d6d-cf2c-cc42-f123a5e3ee11:solaris:excl:-:
```

最初の手順として、現在のホストのゾーンを切り離します。ゾーンは、インストール済み状態から構成済み状態に移ります。すべての ZFS ストレージプールはエクスポートされ、共有ストレージリソースは自動的に構成解除されます。

```
root@initiator:/# zoneadm -z iscsi detach
Exported zone zpool: iscsi_rpool
Unconfigured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E50001
Exported zone zpool: iscsi_data
Unconfigured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E80002
    iscsi://target/lunname.naa.600144F035FF8500000050C884EC0003
```

```
root@initiator:/# zoneadm list -cp
0:global:running:/::solaris:shared:-:none
-:iscsi:configured:/iscsi::solaris:excl:-:
```

現在のホストで必要な最後の手順として、`zonecfg export` を使用してゾーン構成をエクスポートし、作成されたファイルを新しいホストに転送します。

```
root@initiator:/# zonecfg -z iscsi export -f /export/iscsi.cfg
```

新しいホストで、`zonecfg` を使用して、保存したファイルからゾーン構成を最初にインスタンス化します。ゾーンは、構成済みのゾーン状態になります。

```
root@newhost:/# zonecfg -z iscsi -f /export/iscsi.cfg
```

```
root@newhost:/# zoneadm list -cp
0:global:running:/::solaris:shared:-:none
-:iscsi:configured:/iscsi::solaris:excl:-:
```

次に、`zoneadm attach` を使用して、新しいホスト上でゾーンを接続します。ゾーンはインストール済み状態に移ります。すべての共有ストレージリソースが構成され、対応する ZFS ストレージプールが自動的にインポートされます。

```
root@newhost:/# zoneadm -z iscsi attach
Configured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E50001
Imported zone zpool: iscsi_rpool
Configured zone storage resource(s) from:
    iscsi://target/lunname.naa.600144F035FF8500000050C884E80002
    iscsi://target/lunname.naa.600144F035FF8500000050C884EC0003
Imported zone zpool: iscsi_data
Progress being logged to /var/log/zones/zoneadm.20130214T145001Z.iscsi.attach
  Installing: Using existing zone boot environment
    Zone BE root dataset: iscsi_rpool/rpool/ROOT/solaris
    Cache: Using /var/pkg/publisher.
  Updating non-global zone: Linking to image /.
Processing linked: 1/1 done
  Updating non-global zone: Auditing packages.
No updates necessary for this image.

  Updating non-global zone: Zone updated.
    Result: Attach Succeeded.
Log saved in non-global zone as /iscsi/root/var/log/zones/
zoneadm.20130214T145001Z.iscsi.attach
```

```
root@newhost:/# zoneadm list -cp
0:global:running:/::solaris:shared:-:none
-:iscsi:installed:/iscsi:a19fbb45-4af3-670f-c58e-ee48757c75d6:solaris:excl:-:
```

ゾーンは、新しいホストに移行されており、`zoneadm boot` を使用してブートする準備が整っています。

このプロセスは基本的に、Oracle Solaris でサポートされている 3 つの異なるタイプのストレージ URI すべてで同じです。iSCSI ベースまたはファイバチャネルベースのストレージリソースの場合、新しいホストが、同じ論理ユニットおよびターゲットポートにアクセスできることも確認する必要があります。

共有ストレージゾーン構成に対する既存のゾーンの移動

Oracle Solaris では、既存のゾーンインストールを、共有ストレージベースのゾーン構成に変換できます。共有ストレージリソースを使用したインストール済みゾーンを、従来の構成に変換することもできます。プロセスには手動の手順が含まれていることに注意してください。

▼ 共有ストレージ構成に既存のゾーンを移動する方法

既存のインストール済みのゾーンを、`rootzpool` リソースを含む共有ストレージゾーン構成に変換するために、次の手順が必要です。

1. **`zoneadm(1M) shutdown`** コマンドを使用してゾーンを停止します。
2. インストール済みゾーンのアーカイブを作成します。
詳細は、[solaris\(5\)](#) のマニュアルページを参照してください。
3. **`zoneadm(1M) uninstall`** を使用してゾーンをアンインストールします。
この手順は、現在のゾーンインストールをシステムから削除し、ゾーンを構成済みのゾーン状態に戻します。
4. **`zonecfg`** コマンドを使用して、`rootzpool` リソースおよび対応する共有ストレージリソースを、ゾーン構成に追加します。
5. **`zoneadm) install -a`** コマンドを使用して、アーカイブからゾーンを再度インストールします。
このオプションの詳細は [solaris\(5\)](#) を参照してください。
アーカイブベースのインストール中、ゾーンフレームワークは、共有ストレージリソースを構成し、目的の ZFS ストレージプールを作成またはインポートします。
6. これで、**`zoneadm boot`** を使用してゾーンを再度ブートできます。

▼ 共有ストレージ構成から既存のゾーンを移動する方法

インストール済みゾーンをその共有ストレージ構成から移動するには、次の手順を実行します。

1. `zoneadm shutdown` を使用してゾーンを停止します。
2. `rootzpool` リソースをゾーンから削除する場合、インストール済みゾーンのアーカイブを作成します。
詳細は、[solaris\(5\)](#) のマニュアルページを参照してください。
3. `zoneadm detach` を使用してゾーンを切り離します。
ゾーンは構成済みの状態に移ります。ゾーンの切り離し中、ZFS ストレージプールがエクスポートされ、対応するストレージリソースが構成解除されます。
4. `zonecfg` を使用して、目的の `rootzpool` リソースおよび `zpool` リソースをゾーン構成から削除します。
5. 削除されたリソースに応じて、次の手順のいずれかを実行します。
 - `rootzpool` リソースが削除された場合、`zoneadm(1M) install -a` を使用してアーカイブからゾーンを再インストールします。
 - `zpool` リソースだけを削除する場合、`zoneadm(1M) attach` を使用して、ゾーンを再接続するだけです。この手順は、ゾーンを再度インストール済みの状態に戻します。

アーカイブベースのインストールプロセスは、ゾーンインストールを含む、新しいローカル `zonepath` を作成します。

6. 必要に応じて、`zpool` を使用して ZFS ストレージプールを手動で破棄します。

▼ インストール済みゾーンに ZFS ストレージプールを追加する方法

共有ストレージリソースに基づいて、ZFS ストレージプールをインストール済みゾーンに追加できます。すでに存在している ZFS ストレージプールをゾーンに割り当てするには、次の手動の手順を実行します。最初に、`zpool export` コマンドを使用して、既存の ZFS ストレージプールをエクスポートする必要があることに注意してください。

この手順は、ZFS ストレージプールをあるインストール済みゾーンから別のゾーンに移行する場合にも使用できます。これらの手順以外に、最初に `zoneadm detach` を使

用してソースゾーンを切り離し、ZFS ストレージプールを正しくエクスポートする必要があり、`zonecfg` を使用して対応する `zpool` リソースを削除するようにゾーン構成を変更する必要があります。

1. `zoneadm shutdown` を使用してゾーンを停止します。
2. `zoneadm detach` を使用してインストール済みゾーンを切り離します。
ゾーンは構成済みのゾーン状態に移ります。
3. `zonecfg` を使用して、新しい `zpool` リソースおよび対応する共有ストレージリソースをゾーン構成に追加します。
4. `zoneadm attach` を使用してゾーンを再接続します。
ゾーンはインストール済みの状態に戻ります。

接続中、ゾーンフレームワークは、共有ストレージリソースを構成し、既存の ZFS ストレージプールをインポートして、インストール済みゾーンに割り当てます。

5. これで、`zoneadm boot` を使用してゾーンをリブートできます。

参考資料

詳細は、次の資料を参照してください。

マニュアルページ

次のマニュアルページには、共有リソース上で Oracle Solaris ゾーン を構成するために使用するコマンドの情報が記載されています。

- [suri\(5\)](#)
- [suriadm\(1M\)](#)
- [zonecfg\(1M\)](#)
- [zoneadm\(1M\)](#)
- [zones\(5\)](#)
- [zpool\(1M\)](#)
- [itadm\(1M\)](#)
- [stmfadm\(1M\)](#)
- [sasinfo\(1M\)](#)
- [fcinfo\(1M\)](#)

- [solaris\(5\)](#)
- [beadm\(1M\)](#)
- [iser\(7D\)](#)
- [iscsiadm\(1M\)](#)

Oracle Solaris 管理ガイド

Oracle Solaris ドキュメントライブラリ内のガイド『[Oracle Solaris ゾーン構成リソース](#)』は、共有リソース上でのゾーンの構成に関する追加情報を提供しています。

用語集

アカウントイン グの拡張	Solaris オペレーティングシステムで、タスクまたはプロセスに基づくリソース消費量を柔軟に記録できる方法。
完全ルート ゾーン	必要なすべてのシステムソフトウェアおよび追加パッケージがゾーン固有のファイルシステムにインストールされる非大域ゾーンのタイプ。
局所スコープ	制御値を超えようとしているプロセスに対して行われる局所アクション。
公平配分スケ ジューラ	公平さを基準に CPU 時間を割り当てるスケジューリングクラス (FSS と呼ぶ)。配分は、システムの CPU リソースのうちプロジェクトに割り当てる部分を定義します。
作業セットサ イズ	作業セットのサイズ。作業セットとは、プロジェクトの作業負荷がその処理サイクル中にアクティブに使用するページのことです。
作業負荷	アプリケーションまたはアプリケーショングループのプロセスすべての合計。
上限制御	システムリソース使用率を規制するプロセス。
常駐セットサ イズ	常駐セットのサイズ。常駐セットとは、物理メモリーに常駐するページのことです。
スカナ	あまり使用されていないページを識別するカーネルスレッド。メモリーが少ない状況では、スカナは最近使用されていないページを再生します。
静的プール構 成	管理者が、リソースプール機能に関してシステムを構成する方法を表現したもの。
ゾーン管理者	ゾーン管理者の特権は、非大域ゾーンに対してのみ有効です。 大域管理者 も参照してください。
ゾーン状態	非大域ゾーンのステータス。ゾーンの状態は、構成済み、不完全、インストール済み、準備完了、使用不可、稼働、または停止処理のいずれかになります。
大域管理者	root ユーザーまたは root 役割を持つ管理者大域ゾーンにログインすると、大域管理者または適切な承認が付与されたユーザーは、システム全体をモニターしたり制御したりできます。

[ゾーン管理者](#)も参照してください。

大域スコープ	システム上のすべてのリソース制御のリソース制御値に適用されるアクション。
大域ゾーン	すべての Oracle Solaris システムに含まれるゾーン。非大域ゾーンを使用しているときには、大域ゾーンはシステムのデフォルトゾーンであると同時に、システム規模の管理制御に使用されるゾーンでもあります。 非大域ゾーン も参照してください。
タスク	リソース管理において、長時間にわたる作業の集合を表すプロセスの集まり。各タスクは1つのプロジェクトに関連付けられます。
データリンク	OSI プロトコルスタックのレイヤー 2 インタフェース。システム内で STREAMS DLPI (v2) インタフェースとして表されます。このインタフェースは、TCP/IP などのプロトコルスタックで接続できます。Oracle Solaris 10 ゾーンでは、データリンクは物理インタフェース、アグリゲーション、または VLAN でタグ付けされたインタフェースです。データリンクは、物理インタフェースとして参照されることもあります (たとえば、NIC または VNIC を指す場合)。
デフォルトプール	プールが有効に設定される際にシステムにより作成されるプール。 リソースプール も参照してください。
デフォルトプロセッサセット	プールが有効に設定される際にシステムにより作成されるプロセッサセット。 プロセッサセット も参照してください。
動的構成	ある時点における、指定されたシステムのリソースプールフレームワーク内部のリソース配置に関する情報。
動的再構成	SPARC ベースのシステムで、システムの稼働中にハードウェアを再構成する機能。DR とも呼ばれます。
ネームサービスデータベース	このドキュメントのプロジェクトとタスクの概要の章では、LDAP コンテナと NIS マップの両方を指して使用されます。
非大域ゾーン	Oracle Solaris オペレーティングシステムの単一インスタンス内に作成された仮想オペレーティングシステム環境。Oracle Solaris ゾーンソフトウェア区分技術を使用して、オペレーティングシステムサービスが仮想化されます。
非大域ゾーン管理者	ゾーン管理者 を参照してください。
プール	リソースプール を参照してください。

プールデーモン	リソースの動的割り当てが必要な場合にアクティブになる <code>poold</code> システムデーモン。
ブランド	BrandZ 機能のインスタンス。アプリケーションの実行に使用される、ネイティブでないオペレーティング環境が含まれている非大域ゾーンを提供します。
ブランドゾーン	ネイティブでないアプリケーションを非大域ゾーンで実行するための隔離された環境。
プロジェクト	ネットワーク全体の関連作業に対する管理識別子。
プロセッサセット	互いに素である CPU のグループ。各プロセッサセットには、0 以上のプロセッサを含めることができます。プロセッサセットは、リソースプール構成内でリソース要素として表されます。これは <code>pset</code> とも呼ばれます。 素 も参照してください。
メモリー上限 実行しきい値	リソース上限デーモンが上限を制限するときのシステム上における物理メモリーの使用効率 (パーセンテージ)。
素	セットのメンバーが重複しないセットのタイプ。
読み取り専用 ゾーン	読み取り専用ルートを持つように構成された不変ゾーン。
リソース	アプリケーションの動作を変更する目的で操作されるコンピューティングシステムの側面。
リソース管理	アプリケーションが利用可能なシステムリソースをどのように使用するかを制御する機能。
リソース上限 デーモン	リソース上限が定義されたプロジェクト内で動作するプロセスが消費する物理メモリーを規制するデーモン。
リソース消費者	基本的には Solaris のプロセス。プロジェクトやタスクなどのプロセスモデルエンティティにより、集計済みのリソース消費に関して考察できます。
リソース制御	プロセスごと、タスクごと、またはプロジェクトごとのリソース消費量に対する制限。
リソースセット	プロセスをバインド可能なリソース。たいていの場合、カーネルサブシステムにより構築され、ある種の区分化を提供するオブジェクトを指して使用されます。リソースセットの例には、スケジューリングクラスやプロセッサセットが含まれます。
リソースパーティション	リソースの排他的な一部。リソースのパーティションすべての合計は、実行中の単一の Solaris インスタンスで利用可能なリソースの総量を表します。
リソースプール	マシンのリソース区分化に使用する構成メカニズム。リソースプールは、区分化可能なリソースグループ間の関係を表します。

ロックされたメモリー	ページング不可能なメモリー。
cap	システムリソース使用率に対する規制。
FSS	公平配分スケジューラ を参照してください。
Oracle Solaris 10 ゾーン	Oracle Solaris 11 リリースを実行しているシステムの <code>solaris10</code> ブランドゾーンで実行される Solaris 10 アプリケーション用の完全な実行時環境。
Oracle Solaris ゾーン	オペレーティングシステムサービスを仮想化し、アプリケーションを実行するための分離およびセキュリティー保護された環境を提供するソフトウェア区分技術。
RSS	常駐セットサイズ を参照してください。

索引

あ

移行

- zonep2vchk の使用, 93
- 一時的ライブ再構成, 86
- 永続的ライブ再構成, 85, 86

か

クローン

- ZFS, 61
- クローンアーカイブ
作成, 100
- コマンド
ゾーン, 151
- 孤立したブート環境, 97

さ

作成

- クローンアーカイブ, 100
- スナップショット
ZFS, 61
- ゾーン
DTrace の実行, 147
- IPsec, 145
- NFS サーバー, 123
- Oracle Solaris 監査, 146
- solaris、更新, 111
- solaris、パッケージ, 111
- UUID, 53
- zonep2vchk ツール, 93
- zonestat ユーティリティー, 159
- アンインストール, 60
- 移行, 96
- 一時的ゾーン再構成, 85

- 一時的ライブ再構成, 85, 86, 86
- 移動, 63
- インストール, 51
- 永続的ライブ再構成, 85, 86
- クローン, 45, 61
- 検証, 50
- 孤立したブート環境, 97
- サイズの制限, 18
- 削除, 64
- 時間値, 147
- 準備完了状態, 55
- 使用されるコマンド, 151
- 使用できないマシンからの移行, 104
- シングルユーザーモードでブート, 57
- 対話型モード, 76
- 停止, 43, 58
- 停止処理, 43, 57
- ディスク領域, 17
- データの取り込み, 38
- データリンクの管理, 171
- 特権, 141
- 内部構成, 68
- 名前の変更, 31
- ネットワークアドレス, 19
- ネットワーク、共有 IP, 131
- ネットワーク、排他的 IP, 134
- ノード名, 122
- パッケージの削除, 114
- パッケージの追加, 112
- パッケージング, 109
- 非対話型モード, 76
- ブート, 56
- ブート引数, 44, 56
- 不変ゾーン, 179
- 不変大域ゾーン, 183

- プロキシ構成, 114
- ライブ再構成, 85
- リスト, 51
- リポート, 44, 59
- ログインの概要, 67
- ゾーン管理プロファイル, 176
- ゾーンコマンド, 151
- ゾーンコンソールログイン
 - コンソールログインモード, 74
- ゾーンサイズの制限, 18
- ゾーンでの DTrace の実行, 147, 163
- ゾーン内の特権, 141
- ゾーンのアンインストール, 60
- ゾーンの移行, 96
- ゾーンの移動, 63
- ゾーンのインストール, 50, 51
 - 概要, 37
 - タスク, 50
- ゾーンのクローニング, 45, 45, 61
 - ゴールデン, 45
- ゾーンの検証, 50
- ゾーンの削除, 64
- ゾーンの準備完了, 55
- ゾーンの停止, 43, 58
 - トラブルシューティング, 43
- ゾーンの停止処理, 43, 57
- ゾーンの名称の変更, 31
- ゾーンのブート, 56
- ゾーンの並列更新, 116
- ゾーンのホスト名, 18
- ゾーンのリスト, 51
- ゾーンのレポート, 44, 59
- ゾーンへのデータの取り込み, 38
- ゾーンへのログイン
 - フェイルセーフモード, 75
 - リモート, 75

た

- 大域ゾーンのプロキシ, 115
- 大域ゾーンの zone.cpu-shares, 32
- 大域ゾーンプロキシのオーバーライド, 115
- データリンクの管理, 171

な

- ネットワーク、共有 IP, 131
- ネットワーク、排他的 IP, 134
- ノード名
 - ゾーン, 122

は

- ブート引数とゾーン, 56
- 不変ゾーン, 179
 - 管理, 184
- 不変大域ゾーン, 183
- プロキシ構成
 - ゾーン, 114
- プロキシの構成, 114

や

- 読み取り専用ゾーン, 179
 - add dataset ポリシー, 181
 - add fs ポリシー, 181
 - dynamic-zones, 181
 - file-mac-profile, 180
 - 管理, 182
 - 構成, 180
 - ブート, 182
 - ログイン, 183
- 読み取り専用ゾーンの管理, 182
- 読み取り専用ゾーンのブート, 182
- 読み取り専用ゾーンルート, 179, 180

ら

- ライブ再構成, 85
- リモートゾーンへのログイン, 75
- ログイン
 - リモートゾーン, 75

A

- archiveadm コマンド
 - クローンアーカイブの作成, 100

C

create サブコマンド
archiveadm コマンドの例, 100

D

dtrace_proc, 147, 163
dtrace_user, 147, 163

F

force-storage-destroy-all, 198
force-zpool-destroy-all, 198
force-zpool-import, 41
fsstat, 162, 163
fsstatユーティリティー, 122

I

info サブコマンド
archiveadm コマンドの例, 100
IPsec
ゾーンで使用, 145

M

MWAC, 179

N

NFS サーバー, 123

O

Oracle Solaris 監査
ゾーンでの使用, 146

S

solaris ゾーン
手動の同期, 109

Z

ZFS
クローン, 61
スナップショット, 61
zlogin 読み取り専用ゾーン, 183
zoneadm
mark サブコマンド, 54
zoneadm コマンド, 37
zoneadmd デーモン, 41
zonecfg
大域ゾーンでの, 22
手順, 22
zonecfg コマンド, 22
zonep2vchk
移行ツール, 93
zonepath
ZFS に自動的に作成, 51
zonestat, 159
zonestat ユーティリティー, 121
zsched プロセス, 42

