

Oracle® Solaris ゾーン構成リソース

ORACLE®

Part No: E62809
2016年11月

Part No: E62809

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方法	7
1 非大域ゾーンの構成	9
ゾーンのリソースについて	9
ゾーン管理での権利プロファイルと役割の使用	10
zonecfg template プロパティとトークン	11
インストール前の構成処理	13
構成可能なリソースとプロパティ	13
ゾーン名	13
ゾーンパス	14
ゾーンの自動ブート	14
solaris および solaris10 のみ: global-time プロパティ	14
不変ゾーンの file-mac-profile プロパティ	15
admin リソース	15
dedicated-cpu リソース	16
solaris-kz のみ: virtual-cpu リソース	17
capped-cpu リソース	18
スケジューリングクラス	19
capped-memory リソースと物理メモリの制御	19
solaris および solaris10 のみ: npiv リソース	21
solaris および solaris10 のみ: rootzpool リソース	22
zpool リソースの自動追加	24
solaris-kz SPARC のみ: カーネルゾーンの移行クラスとホスト互換性レベル	25
ゾーンネットワークインタフェース	27
ゾーンでマウントされるファイルシステム	33
ファイルシステムのマウントと更新	34
ゾーン内のホスト ID	35
非大域ゾーンの /dev ファイルシステム	35
非大域ゾーンのリムーバブル lofi デバイス	35

非大域ゾーン内でのディスク形式のサポート	36
ストレージ URI を持つカーネルゾーンのデバイスリソース	36
構成可能な特権	38
リソースプールの関連付け	38
ゾーン規模のリソース制御の設定	39
ゾーンのコメントの追加	42
zonecfg コマンドの使用について	42
zonecfg のモード	43
zonecfg の対話型モード	43
zonecfg のコマンドファイルモード	46
ゾーン構成データ	46
リソースタイプとプロパティ	46
リソースタイプのプロパティ	52
Tecla コマンド行編集ライブラリ	66
索引	67

このドキュメントの使用方法

- **概要** – Oracle Solaris ゾーンリソースを構成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。

◆◆◆ 第 1 章

非大域ゾーンの構成

この章では、非大域ゾーン構成で使用する `zonecfg` コマンドのリソースおよびプロパティの概要について説明します。

この章の内容は次のとおりです。

- 9 ページの「ゾーンのリソースについて」
- 13 ページの「インストール前の構成処理」
- 13 ページの「構成可能なリソースとプロパティ」
- 42 ページの「`zonecfg` コマンドの使用について」
- 43 ページの「`zonecfg` のモード」
- 46 ページの「ゾーン構成データ」
- 66 ページの「`Tecla` コマンド行編集ライブラリ」

『Oracle Solaris ゾーンの実装と使用』の第 1 章、「非大域ゾーンの計画および構成方法」に進み、システムにインストールする非大域ゾーンを構成します。

Oracle Solaris 11.3 リリースでは、デフォルトの `solaris` ブランドゾーンは、ネイティブゾーンと呼ばれます。

ゾーンのリソースについて

ゾーン内で制御できるリソースには、次のものがあります。

- リソースプールまたは割り当てられる CPU で、システムリソースの区分に使用されます。
- リソース制御機能。システムリソースに対する制約メカニズムを提供します。
- スケジューリングクラス。使用可能な CPU リソースのゾーン間での割り当てを、それらの重要性に基づいて制御できます。この重要性は、各ゾーンに割り当てる CPU リソースの「配分」で表します。

ゾーン管理での権利プロファイルと役割の使用

root ユーザーにはすべての管理権利があります。root ユーザーは、役割、権利プロファイル、特定の特権や承認など、管理権利をユーザーに割り当てることができます。割り当てられた権利の使用については、『Oracle Solaris 11.3 でのユーザーとプロセスのセキュリティ保護』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。

ゾーンの承認についての詳細は、15 ページの「admin リソース」を参照してください。

非大域ゾーンでの特権の使用については、『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーン内の特権」を参照してください。

ゾーンの権利プロファイルは次のとおりです。

ゾーンのセキュリ ティー	ゾーン仮想アプリケーション環境構成。 ゾーンのセキュリティープロファイルは、ゾーンの管理ではなく、ゾーンの作成にのみ使用されます。プロファイルには承認が含まれます。このプロファイルでは、zoneadm コマンドや zlogin コマンドではなく、zonecfg コマンドを使用できます。
ゾーン構成	ゾーン仮想アプリケーション環境の委任構成。 ゾーン構成プロファイルは、ゾーンの管理を委任するために使用されます。このプロファイルには承認は含まれません。このプロファイルは、admin リソースの auths プロパティが設定されているときに、zonecfg コマンドで使用されます。
ゾーン管理	ゾーン仮想アプリケーション環境の委任管理。 ゾーン管理プロファイルは、ゾーンの管理を委任するために使用されます。このプロファイルには承認は含まれません。このプロファイルは、admin リソースの auths プロパティが設定されているときに、zonecfg コマンドで使用されます。
ゾーンの移行	ゾーン仮想アプリケーション環境の委任移行。 ゾーンの移行プロファイルは、ゾーンの管理を委任するために使用されます。このプロファイルには承認は含まれません。このプロファイルは、admin リソースの auths プロパティが設定されているときに、zonecfg コマンドで使用されます。

プロファイルを使用するには、15 ページの「admin リソース」を参照してください。ゾーンプロファイルについては、profiles(1) と prof_attr(4) のマニュアルページも参照してください。

システム上で実行されているアプリケーションを保護する Oracle Solaris の機能の詳細については、『[Oracle Solaris 11 セキュリティと強化ガイドライン](#)』の「[アプリケーションの保護と分離](#)」を参照してください。

zonecfg template プロパティとトークン

特定のブランドを取得する、空の構成を取得する、またはほかの構成済みのゾーンに類似しているゾーン構成を作成するには、別々のテンプレートを使用します。

次の場合にプロパティを変更するかどうか、およびどのように変更するかを定義するには、zonecfg template プロパティを使用します。

- 新しいリソースインスタンスが構成に追加されたとき。
- 構成のクローニング中、一部のプロパティに一意の値が必要なとき。template プロパティのトークンを使用して、これらの一意の値を指定します。

表 1 zonecfg template のトークン

トークン	説明	使用法
<code>%{zonename}</code>	ゾーンの名前。	zonecfg 内で、ユーザーまたはテンプレート値からの入力として使用できます。
<code>%{id}</code>	リソースの id プロパティ値である一意のインスタンス番号。	zonecfg 内で、ユーザーまたはテンプレート値からの入力として使用できます。特定のリソースの id プロパティに評価されます。id プロパティをサポートするリソース有効範囲内で使用する必要があります。
<code>%{ global-rootzpool}</code>	大域ゾーンのルートプールの名前に評価されます。	デフォルトの solaris-kz device リソースで使用されます。
<code>%%</code>	% に評価されます。	zonecfg 内で、ユーザーからの入力として使用できます。

表 2 リソースプロパティによってサポートされているトークン

リソース	プロパティ	サポートされているトークン
global	zonempath	<code>%{zonename}</code>
dataset	name	<code>%{zonename}</code>
device	match storage	<code>%{zonename}</code> 、 <code>%{id}</code> 、 <code>%{global-rootzpool}</code>

リソース	プロパティ	サポートされているトークン
		<code>{zonename}</code> 、 <code>{id}</code> 、 <code>{global-rootzpool}</code>
fs	dir	<code>{zonename}</code>
net	physical	<code>{id}</code>
anet	linkname	<code>{id}</code>
suspend	storage path	<code>{zonename}</code> 、 <code>{global-rootzpool}</code> <code>{zonename}</code>
rootzpool	storage	<code>{zonename}</code> 、 <code>{global-rootzpool}</code>
zpool	storage	<code>{zonename}</code> 、 <code>{global-rootzpool}</code>

例 1 zonepath での `{zonename}` プロパティ

```
zonecfg:nz> info zonepath
zonepath.template: /system/zones/{zonename}
zonepath: /system/zones/nz
zonecfg:nz> set zonename=new-zone
zonecfg:new-zone> info zonepath
zonepath.template: /system/zones/{zonename}
zonepath: /system/zones/new-zone
zonecfg:new-zone>
```

例 2 solaris-kz device リソースでの storage プロパティに使用されるトークン

```
device 0:
  match not specified
  storage.template: dev:/dev/zvol/dsk/{global-rootzpool}/VARSHARE/zones/{zonename}
/disk{id}
  storage: dev:/dev/zvol/dsk/rpool/VARSHARE/zones/kernel-zone1/disk0
  id: 0
  bootpri: 0
```

注記 - SYSsolaris-kz テンプレートを使用すると、solaris-kz ブランドゾーンを構成できます。デフォルトでは、SYSsolaris-kz テンプレートは、4つの仮想 CPU と 4G バイトのメモリーを持つゾーンを構成します。

SYSsolaris-kz-minimal テンプレートを使用すると、最小限のカーネルゾーンを構成できます。SYSsolaris-kz の最小限のテンプレートでは、1つの仮想 CPU と 2G バイトのメモリーを持つゾーンが構成されます。

ゾーンのリモート管理デーモン (RAD) のモジュール構成は、プロパティテンプレートを使用することで、変更を表現、適用、または実装する体系的な方法を提供します。zonemgr(3RAD) のマニュアルページを参照してください。最初に rad-zonemgr

パッケージがシステムにインストールされておらず、あとで `pkg install` を使用してインストールした場合、`rad:local` を再起動する必要があります。また、`rad:remote` が実行されていた場合は、これも再起動します。再起動するには、`svcadm(1M)` を使用します。RAD デーモンがモジュールをロードしたことを確認します。

インストール前の構成処理

非大域ゾーンをインストールしてシステムで使用する前に、ゾーンを構成する必要があります。

`zonecfg` コマンドを使用すると、構成を作成したり、指定されたリソースやプロパティが仮定のシステム上で有効かどうかを判定したりできます。特定の構成について `zonecfg` で実行される検査では、次のことが確認されます。

- `solaris` および `solaris10` ブランドゾーンの場合、ゾーンパスが指定されていること。
- 各リソースの必須プロパティがすべて指定されていること。
- 構成に矛盾がないこと。たとえば、`anet` リソースが存在する場合、ゾーンは排他的 IP タイプであり、共有 IP ゾーンにはできません。また、別名が付いたデータセットでデバイスに関する潜在的競合が存在する場合、`zonecfg` コマンドは警告を出力します。

`zonecfg` コマンドの詳細は、[zonecfg\(1M\)](#) のマニュアルページを参照してください。

構成可能なリソースとプロパティ

このセクションでは、構成可能な必須およびオプションのゾーンリソースおよびプロパティについて説明します。ゾーン名だけが必要です。詳細は、[46 ページの「ゾーン構成データ」](#) を参照してください。特定ブランドのゾーンに固有の構成オプションの詳細は、[solaris\(5\)](#) および [solaris-kz\(5\)](#) のマニュアルページを参照してください。

ゾーン名

ゾーンの名前を選択する必要があります。

ゾーンパス

パスを指定しない場合、`zonepath` のデフォルト値は `/system/zones/{zonename}` です。ゾーン構成に `rootzpool` リソースがない場合は、ZFS データセット `{global-rootzpool}/VARSHARE/system/zones/{zonename}` が作成され、`/system/zones/{zonename}` にマウントされます。

ゾーンのパスを選択する場合、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は、データセットでもなければならぬことに注意してください。`zonepath` データセットが自動的に作成されない場合のみ、`zonepath` の親が ZFS データセットでなければなりません。

カーネルゾーンでは `zonepath` プロパティはサポートされません。ゾーンルートは ZFS ボリューム内に保持されます。ゾーンのインストール先のデバイスは、任意の正の整数値に設定された `bootpri` プロパティを持つ `device` リソースで指定されます。

ゾーンの自動ブート

`autoboot` プロパティの設定は、大域ゾーンのブート時にゾーンが自動的にブートされるかどうかを決定します。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。

solaris および solaris10 のみ: global-time プロパティ

非大域ゾーン内から、ゾーン固有の時間とシステム全体の時間のどちらの変更を許可するかを指定するには、`global-time` プロパティを設定します。

- `global-time` プロパティの `global-time=true` の値は、システム全体の時間をゾーンで設定できることを示します。
- `global-time` プロパティの `global-time=false` の値は、ゾーン固有の時間をゾーンで設定できることを示します。

例 3 ゾーンでのゾーン固有の時間の設定の有効化

```
# zonecfg -z my-zone
zonecfg:my-zone> set global-time=false
zonecfg:my-zone> exit
```

`global-time` プロパティの値を割り当てる必要があります。ただし、値が設定されていなくても、`limitpriv` プロパティを使用して `sys_time` 特権が明示的に割り当てられている場合は、`global-time` の値は `true` として扱われます。`limitpriv` プロパティを使用して `sys_time` 特権が明示的に割り当てられていなければ、`global-time` は `false` として扱われます。

Oracle Solaris 11.3 では、`global-time` プロパティ設定に応じて、非大域ゾーン内のプロセスが、次のシステムコールによって仮想ゾーン固有の時間かシステム全体の時間のどちらかを操作できます。プロセスに `sys_time` 特権が割り当てられている必要があります。

- `stime(2)`
- `clock_settime(3C)`
- 時刻クロックを書き込む IA 固有のリアルタイムクロック (RTC) コール

詳細は、『Oracle Solaris ゾーンの実成と使用』の「非大域ゾーン内の特権」を参照してください。

不変ゾーンの `file-mac-profile` プロパティ

`file-mac-profile` を使用して、読み取り専用ルートで不変ゾーンを構成します。

詳細は、『Oracle Solaris ゾーンの実成と使用』の第 11 章、「不変ゾーンの構成と管理」を参照してください。

admin リソース

`admin` 設定を使用すると、ゾーン管理の承認を設定できます。承認を定義するための推奨の方法は、`zonecfg` コマンドによる方法です。

`user` ユーザー名を指定します。

`auths` ユーザー名に対して承認を指定します。

`auths` に使用できる値は次のとおりです。

`solaris.zone.login` RBAC を使用中の場合は、このゾーンへの `zlogin` を認証済みで使用許可します。対話型ログインには `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。詳細は、`zlogin(1)` および『Oracle Solaris ゾーンの実成と使用』の第 4 章、「非大域ゾーンへのログインについて」を参照してください。

<code>solaris.zone.manage</code>	RBAC を使用中の場合は、構成されたゾーンの通常の管理を許可します。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、 <code>solaris.zone.manage/zonename</code> の承認が必要です。
<code>solaris.zone.clonefrom</code>	RBAC を使用中の場合は、新しいゾーンのクローンを作成するソースとして、指定のゾーンを使用できるようにします。別のゾーンのコピーを作成するサブコマンドは <code>solaris.zone.clonefrom/source_zone</code> の承認を必要とします。
<code>solaris.zone.config</code>	RBAC を使用中の場合は、 <code>solaris.zone.config/zonename</code> の承認を使用して、ゾーンの永続的な構成の変更を許可します。永続的な構成の詳細は、『Oracle Solaris ゾーンの作成と使用』の第6章、「ライブゾーン再構成」を参照してください。
<code>solaris.zone.liveconfig</code>	RBAC を使用中の場合は、 <code>solaris.zone.liveconfig/zonename</code> の承認を使用して、ライブゾーン構成の検査および変更を許可します。ライブゾーン構成の詳細は、『Oracle Solaris ゾーンの作成と使用』の第6章、「ライブゾーン再構成」を参照してください。

承認の詳細は、`auths(1)`、`auth_attr(4)`、および `user_attr(4)` を参照してください。

dedicated-cpu リソース

`dedicated-cpu` リソースを使用して、非大域ゾーンの実行中に、システムのプロセッサの一部をそのゾーン専用として指定します。ゾーンのブート時に、ゾーンの実行中に使用される一時プールが動的に作成されます。

`zonecfg` で指定すると、移行時にプールの設定が伝達されます。

`dedicated-cpu` リソースは、`ncpus` の制限を設定し、必要に応じて `importance` も設定します。

<code>ncpus</code>	<p>CPU の数を指定するか、CPU の数の範囲を 2-4 などと指定します。リソースプールの動的な動作を得るために範囲を指定する場合は、次の手順も実行してください。</p> <ul style="list-style-type: none"> ■ <code>importance</code> プロパティを設定します。 ■ <code>poold</code> サービスを有効にします。手順については、『Oracle Solaris 11.3 でのリソースの管理』の「<code>svcadm</code> を使用して動的リソースプールサービスを有効にする方法」を参照してください。
<code>importance</code>	動的な動作のために CPU 範囲を使用している場合は、 <code>importance</code> プロパティも設定します。 <code>importance</code> プロ

パーティはオプションですが、プールの相対的な重要性を定義します。このプロパティが必要となるのは、ncpus に範囲を指定した場合で、poolid によって管理される動的リソースプールを使用しているときだけです。poolid が実行されていない場合、importance は無視されます。poolid が実行されている場合、importance が設定されていないと、importance のデフォルトは 1 になります。詳細は、『Oracle Solaris 11.3 でのリソースの管理』の「pool.importance プロパティの制約」を参照してください。

次のプロパティを使用して、cpus、cores、および sockets に永続的な dedicated-cpu リソースを設定します。

cpus	特定の CPU を永続的にゾーンに割り当てます。
cores	特定のコアを永続的にゾーンに割り当てます。
sockets	指定された数のソケットを永続的に割り当てます。

システムのリポートごとの結果の不整合をなくすには、dedicated-cpu:cpus を使用して、使用する正確な CPU を指定します。ncpus だけを指定する自動 virtual-cpu リソースではなく、dedicated-cpu リソースを使用します。

注記 - capped-cpu リソースと dedicated-cpu リソースには互換性はありません。cpu-shares リソース制御と dedicated-cpu リソースには互換性はありません。

注記 - 使用可能な CPU の数に自動的にサイズ設定して拡大縮小するアプリケーションが capped-cpu 制限を認識しないことがあります。すべての CPU が使用可能として認識されると、Oracle データベースや Java 仮想マシン (JVM) などのアプリケーションでスケーリングとパフォーマンスに影響を与える可能性があります。アプリケーションが動作していないか、使用できないように見えることがあります。パフォーマンスが重要である場合は、capped-cpu で JVM を使用しないでください。影響を受けるカテゴリ内のアプリケーションでは dedicated-cpu リソースを使用できます。

solaris-kz のみ: virtual-cpu リソース

カーネルゾーンの仮想 CPU (VCPU) にデフォルト以外の数を割り当てる場合は、virtual-cpu リソースを使用して VCPU の数を設定します。

デフォルトのカーネルゾーン構成の VCPU は 4 つです。各 virtual-cpu は、最大 1 ホスト CPU の演算能力を使用できますが、システム CPU リソースの競合が起きると低下する場合があります。カーネルゾーンに割り当てられる CPU は、ncpus 値によって定義されます。virtual-cpu プロパティを追加すると、さらに CPU をカーネルゾーンに追加できます。

カーネルゾーンが `dedicated-cpu` または `pool` リソースを使用して作成されたプール内にある場合、作成される仮想 CPU の数はそのプールのサイズに適合します。VCPU サイズは FSS 配分の数に基づいていないことに注意してください。

CPU リソースが多数のコンシューマ間で共有されている場合、システムによってカーネルゾーンのすべてまたは一部が「スケジュール解除」される期間が存在することがあります。

窃取時間は、システムが CPU リソースを使用している可能性があるためにカーネルゾーンが実行できない時間を示します。

CPU アカウンティング状態 `CMS_STOLEN` は、CPU がこの状態にある時間を示します。この時間は、物理ハードウェア上で実行しているシステムでは、常に 0 です。カーネルゾーンの一部として実行している CPU の場合、ゼロ以外のこの状態の値は、仮想 CPU が実際には物理 CPU にアクセスできなかった時間を反映しています。窃取時間は、`zonestat(1)`、`mpstat(1M)`、`iostat(1M)`、`vmstat(1M)`、およびその他のユーティリティーによって報告されます。

`dedicated-cpu` リソースがすでに定義されている場合、仮想プラットフォームで構成されている仮想 CPU のデフォルト数は、`dedicated-cpu` リソース内の `ncpus` の範囲の下限値と一致します。`dedicated-cpu` と `virtual-cpu` のリソースを両方とも設定する必要はありません。

capped-cpu リソース

`capped-cpu` リソースは、1 つのプロジェクトまたは 1 つのゾーンで消費可能な CPU リソース量に対して絶対的できめの細かい制限を設けます。プロセッサセットと組み合わせると、CPU キャップはセット内の CPU 使用率を制限します。`capped-cpu` リソースには、小数点第 2 位までの正の小数である 1 つの `ncpus` プロパティがあります。このプロパティは、CPU のユニット数に対応しています。このリソースには範囲を指定できません。このリソースには小数を指定できます。`ncpus` を指定する場合、1 の値は 1 つの CPU の 100% を意味します。1.25 の値は 125% を意味します。100% がシステム上の 1 つの CPU の上限となります。

注記 - `capped-cpu` リソースと `dedicated-cpu` リソースには互換性がありません。

注記 - 使用可能な CPU の数に自動サイズ設定して自動的に拡大縮小するアプリケーションが `capped-cpu` 制限を認識しないことがあります。すべての CPU が使用可能として認識されると、Oracle データベースや Java 仮想マシン (JVM) などのアプリケーションでスケーリングとパフォーマンスに影響を与える可能性があります。アプリケーションが動作していないか、使用できないように見えることがあります。パフォーマンスが重要である場合は、`capped-cpu` で JVM を使用しないでください。影響を受けるカテゴリ内のアプリケーションでは `dedicated-cpu` リソースを使用できません。16 ページの「[dedicated-cpu リソース](#)」を参照してください。

スケジューリングクラス

公平配分スケジューラ (FSS) では、各ゾーンに対する使用可能な CPU リソースの割り当てを、そのゾーンの重要性に基づいて制御します。この重要性は、各ゾーンに割り当てる CPU リソースの「配分」で表します。CPU リソースのゾーン間での割り当てを管理するために FSS を使用していない場合でも、ゾーン内のプロジェクトに配分を設定するために FSS を使用するよう、ゾーンのスケジューリングクラスを設定することができます。

`cpu-shares` プロパティを明示的に設定すると、公平配分スケジューラ (FSS) はそのゾーンのスケジューリングクラスとして使用されます。ただし、この場合に望ましい FSS の使用法は、`dispadm` コマンドを使用して、FSS をシステムのデフォルトのスケジューリングクラスに設定する方法です。このようにすると、すべてのゾーンがシステムの CPU リソースの公平配分を受けることができます。ゾーンに対して `cpu-shares` が設定されていない場合、そのゾーンはシステムのデフォルトのスケジューリングクラスを使用します。ゾーンのスケジューリングクラスは、次の処理によって設定されます。

- `zonecfg` の `scheduling-class` プロパティを使ってゾーンのスケジューリングクラスを設定できます。
- リソースプール機能を使ってゾーンのスケジューリングクラスを設定できます。ゾーンがプールに関連付けられている場合、そのプールの `pool.scheduler` プロパティに有効なスケジューリングクラスが設定されていると、ゾーンで実行されるプロセスは、デフォルトでそのスケジューリングクラスで実行されます。『Oracle Solaris 11.3 でのリソースの管理』の「リソースプールの紹介」および『Oracle Solaris 11.3 でのリソースの管理』の「プールをスケジューリングクラスに対応付ける方法」を参照してください。
- `cpu-shares` リソース制御が設定されている場合で、別のアクションを通じて FSS がゾーンのスケジューリングクラスとして設定されていないときは、ゾーンのブート時に `zoneadm` によってスケジューリングクラスが FSS に設定されます。
- ほかの処理を通してスケジューリングクラスが設定されていない場合、ゾーンはシステムのデフォルトのスケジューリングクラスを継承します。

`pricntl` コマンド (`pricntl(1)` のマニュアルページに記載) を使用すると、デフォルトのスケジューリングクラスの変更やリブートを行うことなく、実行中のプロセスを別のスケジューリングクラスに移動できます。

capped-memory リソースと物理メモリの制御

`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。46 ページの「リソースタイプとプロパティ」の `capped-memory` も参照してください。

solaris ゾーンと capped-memory リソース

ネイティブ (solaris) ブランドゾーンの場合、capped-memory リソースは、physical、swap、および locked メモリープロパティの制限を設定します。各制限はオプションですが、少なくとも 1 つは設定する必要があります。

- 大域ゾーンから rcapd を使用してネイティブゾーンのメモリー上限を設定する場合は、physical プロパティの値を決定します。capped-memory リソースの physical プロパティは、ゾーンの max-rss 値として rcapd で使用されます。
capped-memory リソースの physical プロパティは、rcapd によって強制される弱い RAM 割り当て制限を表します。ゾーンが物理的な限度に達しても、そのゾーンは RAM の割り当てを続行できますが、システムで全体的なメモリー不足が起きていない場合でも、スワップデバイスへのページングが行われます。ページングは大量の I/O を生じさせる場合があるため、システム上のほかの操作に悪影響を与える可能性があります。一方、スワップは制限しても、システムのページングアクティビティに直接の影響はありません。physical を設定せずに swap を設定すると、ネイティブゾーンで使用されるメモリーの量を効果的に制限できます。
- ゾーンによって割り当て可能なスワップの量を制限するときには、ゾーンによって割り当て可能な RAM の量も制限します。ゾーンは、スワップよりも多くの RAM を割り当ててはできません。ゾーンがそのスワップ限度に達すると、システムで全体的なメモリー不足が起きていない場合でも、ゾーンでの新規メモリー割り当ては失敗します。
capped-memory リソースの swap プロパティは、ネイティブゾーンの zone.max-swap リソース制御を設定するための望ましい方法です。
- capped-memory リソースの locked プロパティは、ネイティブゾーンの zone.max-locked-memory リソース制御を設定するための望ましい方法です。

注記 - 通常はアプリケーションが多量のメモリーをロックすることはありませんが、ゾーンのアプリケーションによってメモリーがロックされることがわかっている場合は、ロックされるメモリーを設定するとよいでしょう。ゾーンの信頼が問題になる場合は、ロックされるメモリーの上限を、システムの物理メモリーの 10 パーセントまたはゾーンの物理メモリー上限の 10 パーセントに設定することもできます。

詳細は、『Oracle Solaris 11.3 でのリソースの管理』の第 10 章、「リソース上限デーモン (rcapd) による物理メモリーの制御について」、『Oracle Solaris 11.3 でのリソースの管理』の第 11 章、「リソース上限デーモンの管理のタスク」、および『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」を参照してください。ゾーンのリソース上限を一時的に設定するには、『Oracle Solaris 11.3 でのリソースの管理』の「ゾーンに一時的なリソース上限を指定する方法」を参照してください。

solaris-kz ゾーンと capped-memory リソース

カーネルゾーンの場合、`physical` プロパティは必須です。`physical` は、カーネルゾーンのメモリー用に予約された RAM の容量を表します。`physical` プロパティを指定する場合は、物理メモリーでラージページを使用するためのポリシーを設定する、`pagesize-policy` プロパティも指定するようにしてください。

`physical` に設定された値によって、ブートが成功するかどうかが決まります。物理マシンが 1G ページをサポートしている場合でも `physical` が 2M に設定されていると、ホストはより大きいページサイズをサポートしていますが、`physical` の値が使用されます。したがって、`pagesize-policy=largest-available` を使用したブートは成功します。

物理マシン上で使用可能なページサイズを判別するには、次のように入力します。

```
user% pagesize -a
```

デフォルトの `solaris-kz` テンプレートは、`pagesize-policy` を推奨値である `largest-available` に設定します。カーネルゾーンでは、`swap` および `locked` の制限は許可されません。`rcapd` ユーティリティーは使用されません。ライブゾーン再構成はサポートされていません。

プロパティのシステムサポートを判別するには、次のコマンドを使用します。

```
# zoneadm migrate -n
```

`pagesize-policy` プロパティをサポートしていない Oracle Solaris システムは、デフォルト値として `compatible` を使用します。このポリシーをクリアする必要があるのは、古い Oracle Solaris インスタンスからライブ移行する場合と、古い Oracle Solaris インスタンスでカーネルゾーンを再開する場合だけです。

`pagesize-policy` プロパティの値は、[52 ページの「リソースタイプのプロパティ」](#) で説明されています。

詳細については、[solaris-kz\(5\)](#) のマニュアルページも参照してください。

solaris および solaris10 のみ: npiv リソース

`npiv` リソースは、Oracle Solaris ゾーンおよび Oracle Solaris 10 ゾーンの `N_Port_ID Virtualization (NPIV)` をサポートしています。`npiv` リソースは、ゾーンのルートファイルシステムに対するバックエンドストレージとしてファイバチャネルデバイスを持ち、データに対してほかのデバイスを使用するゾーンを構成するために使用されます。

次の例では、2つの `npiv` リソースをゾーン `my-zone` に委任します。`virtual-port-wwn` と `over-hba` はどちらもオプションです。2つの `npiv` ポートが、ゾーンのインストール時に自動的に作成されます。

```
zonecfg:my-zone> add npiv
zonecfg:my-zone:npiv> set virtual-port-wwn=2100000000000001
zonecfg:my-zone:npiv> set over-hba=c9
zonecfg:my-zone:npiv> end
zonecfg:my-zone> add npiv
zonecfg:my-zone:npiv> end
zonecfg:my-zone:>
```

NPIV ポートから認識できるディスクは、ゾーン内部でも認識できます。ファブリックに追加されたディスクは、自動的にゾーン内から認識できます。ファブリックから削除されたディスクは、自動的にゾーンビューから削除されます。

`virtual-port-wwn` プロパティタイプは `npiv` リソースタイプではオプションです。作成される `npiv` のポート World Wide Name (PWWN) が含まれます。ユーザーが指定しない場合、ポートは自動的に生成されます。デフォルトの `virtual-port-wwn` プロパティ値をオーバーライドするには、`npiv` リソーススコープ内から次のコマンドを使用します。

```
zonecfg:my-zone:npiv> set virtual-port-wwn=World Wide Name
```

`zonecfg` コマンドは、文字列が有効であることを確認します。

solaris および solaris10 のみ: rootzpool リソース

`zonecfg` ユーティリティのオプションの `rootzpool` リソースは、`solaris` および `solaris10` ブランドゾーンのゾーンインストール用に専用の `zpool` を作成するために使用されます。ゾーンのルート `zpool` は、1 つ以上の URI (Universal Resource Identifier) によって定義された共有ストレージデバイスでホストできます。必須の `storage` プロパティは、ゾーンのルート `zfs` ファイルシステムを格納するストレージオブジェクトの URI を識別します。特定のゾーンに対して定義できる `rootzpool` は 1 つだけです。このストレージは、ゾーンがブートされたときに、そのゾーンのために自動的に構成されます。

対応する `zpool` は、ゾーンのインストールまたはゾーンの接続操作中に自動的に作成またはインポートされます。`rootzpool` と `zpool` の両方のリソースで、ゾーンがインストールされるとすぐに `zpool` ミラーを自動的に作成できます。詳細は、『[Oracle Solaris ゾーン](#)の作成と使用』の第 13 章、「[共有ストレージでの Oracle Solaris ゾーンの使用開始](#)」を参照してください。

ゾーンのアンインストールまたは切り離しを行うと、次のアクションが実行されます。

- 対応する `zpool` が自動的にエクスポートまたは破棄されます。
- ストレージリソースが自動的に構成解除されます。

事前作成済みの zpool をゾーンのインストールに再利用するには、zpool をシステムからエクスポートする必要があります。

ゾーンのフレームワークでは、次の URI タイプがサポートされています。

- **dev**

ローカルデバイスパスの URI

形式:

```
dev:local-path-under-/dev
dev://absolute-path-with-dev
dev:absolute-path-with-dev
```

例:

```
dev:dsk/c7t0d0s0
dev:///dev/dsk/c7t0d0s0
dev:/dev/dsk/c7t0d0s0
dev:chassis/SYS/HD1/disk
```

- **lu (論理ユニット)**

ファイバチャネル (FC) および Serial Attached SCSI (SAS)

形式:

```
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
```

例:

```
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
```

- **iscsi**

iSCSI の URI

形式:

```
iscsi:///luname.naa.ID
iscsi:///luname.eui.ID
iscsi://host[:port]/luname.naa.ID
iscsi://host[:port]/luname.eui.ID
iscsi:///target.IQN,lun.LUN
iscsi://host[:port]/target.IQN,lun.LUN
```

例:

```
iscsi:///lunname.eui.0021280001cf80f6
iscsi:///lunname.naa.600144f03d70c80000004ea57da10001
iscsi://[::1]/lunname.naa.600144f03d70c80000004ea57da10001
iscsi://127.0.0.1/lunname.naa.600144f03d70c80000004ea57da10001
iscsi://hostname:1234/lunname.eui.0021280001cf80f6
iscsi://hostname:3260/lunname.naa.600144f03d70c80000004ea57da10001
```

```
iscsi://127.0.0.1/target.iqn.com.sun:02:d0f2d311-f703,lun.0
iscsi:///target.iqn.com.sun:02:d0f2d311-f703,lun.6
iscsi://[::1]:1234/target.iqn.com.sun:02:d0f2d311-f703,lun.2
iscsi://hostname:1234/target.iqn.com.sun:4db41b76-e3d7-cd2f-bf2d-9abef784d76c,lun.0
```

ストレージの URI に基づいて共有オブジェクトを管理するには、`suriadm` ツールを使用します。ID、NAA (Name Address Authority)、および既存のストレージオブジェクトの URI の取得については、[suriadm\(1M\)](#) および [suri\(5\)](#) のマニュアルページを参照してください。

システムは、関連するゾーンに対して新規作成またはインポートされた `rootzpool` に名前を割り当てます。割り当てられる名前は、`zonename_rpool` という形式になります。

`storage` プロパティは、`rootzpool` リソーススコープ内から次のコマンドを使用して管理されます。

- `add storage URI string`
- `remove storage URI string`

zpool リソースの自動追加

`zpool` を非大域ゾーンに委任するには、`zonecfg` ユーティリティーでオプションの `zpool` リソースを構成します。`zpool` は、ゾーンがブートされたときに、そのゾーンのために自動的に構成されます。

対応する `zpool` は、ゾーンのインストールまたはゾーンの接続操作中に自動的に作成またはインポートされます。

ゾーンのアンインストールまたは切り離しを行うと、次のアクションが実行されます。

- 対応する `zpool` が自動的にエクスポートまたは破棄されます。
- ストレージリソースが自動的に構成解除されます。

必須の `storage` プロパティは、このリソースに関連付けられたストレージオブジェクトの URI を識別します。

storage プロパティは、zpool リソーススコープ内の次の設定を使用して管理されます。

- add storage *URI string*
- remove storage *URI string*

zpool リソースには name プロパティが必須です。このプロパティは、ゾーンに委任される zpool の名前で使用されます。ZFS ファイルシステムの name コンポーネントにスラッシュ (/) を含めることはできません。

新規作成またはインポートされた zpool に割り当てられる名前は、name プロパティの値です。これは、非大域ゾーン内で表示可能な zpool 名です。新規作成またはインポートされた zpool 名に割り当てられる名前は、大域ゾーンから表示される際には *zonename_name* という形式になります。

注記 - 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ている場合は、ゾーンのインストールが失敗することがあります。詳細は、『Oracle Solaris ゾーンの実装と使用』の「構成済みのゾーンをインストールする方法」の手順 4 を参照してください。

solaris-kz SPARC のみ: カーネルゾーンの移行クラスとホスト互換性レベル

カーネルゾーンから見えるのは、移行クラスとホスト互換性レベルの両方によって有効になっている機能だけです。カーネルゾーンを移行するには、カーネルゾーンから見える機能がソースとターゲットの両方のホストで同じになるように、移行クラスの *cpu-arch* および *host-compatible* プロパティを構成する必要があります。

設定されていない場合は、*cpu-arch* のデフォルト値は *native* です。ゾーンは、ホストと同じ CPU クラスでブートします。ホストの CPU クラスと互換性のある CPU タイプ間でゾーンを移行できます。デフォルトでは、カーネルゾーンの SSM (Silicon Secured Memory) (ADI とも呼ばれる) はオフになっています。

solaris-kz SPARC のみ: CPU 間の移行

特定の CPU クラスでカーネルゾーンを構成するには、*cpu-arch* グローバルプロパティを使用します。CPU クラスは、ホスト CPU クラスとは無関係にできるため、異なる CPU タイプ間で安全に移行できます。特定のクラスを使用して Oracle VM Server for SPARC ゲストドメインをブートする場合、互換性のある CPU タイプのすべてのプラットフォーム間で、ゲストを安全に移行できます。カーネルゾーンは、ゲストドメインと同じ CPU クラスのセットを使用します。

設定されていない場合は、`cpu-arch` のデフォルト値は `native` です。ゾーンは、ホストと同じ CPU クラスでブートします。ホストの CPU クラスと互換性のある CPU タイプ間でゾーンを移行できます。

互換性のないプラットフォームでは、ホストはこれまでに中断したゾーンを再開しません。また、ホストのプラットフォームに対して移行クラスが互換性のない値に設定されている場合も、ホストはゾーンをブートしません。たとえば、`cpu-arch` が `sparc64-class1` に設定されている場合、T5 上のゲストはブートしません。ゾーンの CPU クラスは、ホストの CPU クラスの限度を超えることはできません。

`generic` クラスでブートしたカーネルゾーンは、SPARC T4 以前のシステムに移行できません。カーネルゾーンは、SPARC T4 と Fujitsu M10、または SPARC M10 以降のサポート対象のシステム上で実行します。

```
cpu-arch=generic|migration-class1|sparc64-class1
```

値は次のとおりです。

<code>generic</code>	カーネルゾーンは、T4 よりも新しいシステム間で、CPU タイプとは関係なく移行を実行できます。
<code>migration-class1</code>	カーネルゾーンは SPARC T4、SPARC T5、SPARC T7、SPARC S7、SPARC M5、SPARC M6、および SPARC M7 の間で CPU タイプ間移行を実行できます。
<code>sparc64-class1</code>	カーネルゾーンは Fujitsu M10 と SPARC M10 の間で、CPU タイプ間移行を実行できます。

`cpu-arch` プロパティの設定および確認:

```
# zonecfg -z vz1
zonecfg:vz1> info cpu-arch
cpu-arch: generic
zonecfg:vz1> set cpu-arch=migration-class1
zonecfg:vz1> info cpu-arch
cpu-arch: migration-class1
zonecfg:vz1>
exit
```

solaris-kz SPARC のみ: `host-compatible` プロパティ

SSM (Silicon Secured Memory) 機能 (ADI とも呼ばれる) を有効にするには、`host-compatible` プロパティ `adi` を使用します。デフォルトでは、カーネルゾーンの SSM はオフになっています。SSM を有効にするには、`host-compatible` 修飾子を設定する必要があります。大域ゾーンでは、SSM 対応のハードウェア上で SSM は常にオンになります。

値が設定されていない場合、カーネルゾーンのデフォルトのホスト互換性レベルには、Oracle Solaris 11.2 リリースでサポートされている機能のみが含まれます。

現在のバージョンの Oracle Solaris にある SSM などのすべての機能をサポートするには、`native` ホスト互換性レベルを設定します。`native` ホスト互換性レベルでは、別のリリースの Oracle Solaris を実行しているホストにカーネルゾーンを移行できない場合があることに注意してください。

```
host-compatible=native
```

SSM 機能を有効にするには、`adi` 修飾子を設定します。`adi` 修飾子はデフォルトの互換性レベルでのみ使用できます。

```
host-compatible=adi
```

移行クラスで SSM 機能がサポートされていない場合、`host-compatible` 修飾子を使用して SSM を有効にすることはできません。

ゾーンネットワークインタフェース

ネットワーク接続を提供するために `zonecfg` ユーティリティーによって構成されるゾーンネットワークインタフェースは、ゾーンの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。

インターネットプロトコル (IP) 層は、ネットワークのパケットの受信と配信を行います。この層には、IP ルーティング、アドレス解決プロトコル (ARP)、IP セキュリティーアーキテクチャー (IPsec)、および IP フィルタが含まれます。

非大域ゾーンに使用できる IP タイプには、共有 IP と排他的 IP の 2 種類があります。排他的 IP がデフォルトの IP タイプです。共有 IP ゾーンは、大域ゾーンとネットワークインタフェースを共有します。大域ゾーン内の構成は、共有 IP ゾーンを使用するために `ipadm` ユーティリティーによって実行する必要があります。排他的 IP ゾーンには専用のネットワークインタフェースが必要です。`anet` リソースを使用して排他的 IP ゾーンを構成すると、専用の VNIC が自動的に作成され、そのゾーンに割り当てられます。自動化された `anet` リソースを使用することによって、大域ゾーン内にデータリンクを作成および構成し、非大域ゾーンにデータリンクを割り当てる必要がなくなります。次のことを行うには、`anet` リソースを使用します。

- 大域ゾーンの管理者が、非大域ゾーンに割り当てられたデータリンク用の特定の名前を選択できるようにする
- 複数のゾーンで同じ名前のデータリンクを使用できるようにする

あるアドレスを自動的に構成する必要があり、ほかのアドレスはゾーン内でオンラインおよびオフラインにできる必要がある場合、複数の `anet` リソースを使用できます。たとえば、次の構成には、2 つの `anet` リソースがあります。最初のリソースは、ゾーンのいずれかのインタフェースで `192.168.3.3` を自動的に構成します。2

番目のリソースでは、ゾーンは、別のインタフェース上で 192.168.3.100 および 192.168.3.101 のみを構成できます。

```
zonecfg:t> select anet linkname=net0
zonecfg:t:anet> set allowed-address=192.168.3.3/24
zonecfg:t:anet> set configure-allowed-address=true
zonecfg:t:anet> end
zonecfg:t> add anet
zonecfg:t:anet> set allowed-address=192.168.3.100/24,192.168.3.101/24
zonecfg:t:anet> set configure-allowed-address=false
zonecfg:t:anet> end
zonecfg:t>
```

下位互換性を維持するために、事前構成済みのデータリンクを非大域ゾーンに割り当てることができます。

各タイプの IP 機能については、『Oracle Solaris ゾーンの作成と使用』の「排他的 IP 非大域ゾーンにおけるネットワーク」および『Oracle Solaris ゾーンの作成と使用』の「共有 IP 非大域ゾーンにおけるネットワーク」を参照してください。

注記 - ゾーンを実行するシステムで、『Oracle Solaris 11.3 でのネットワークのセキュリティ保護』で説明されているリンク保護を使用できます。この機能は、大域ゾーン内で構成されます。

データリンクについて

データリンクとは、OSI プロトコルスタックのレイヤー 2 物理インタフェースのことです。このインタフェースは、システム内で STREAMS DLPI (v2) インタフェースとして表されます。このようなインタフェースは、TCP/IP などのプロトコルスタックの下で接続できます。データリンクは、「物理インタフェース」(たとえば、ネットワークインタフェースカード (NIC)) とも呼ばれます。データリンクは、zonecfg(1M) を使用して構成される physical プロパティです。physical プロパティは VNIC でもかまいません。

Oracle Solaris 11 のデフォルトでは、物理ネットワークデバイス名は、nxge0 などのデバイスドライバ名ではなく、net0 などの総称名を使用します。

ゾーンでの IPoIB (IP over Infiniband) の使用については、52 ページの「リソースタイプのプロパティ」の anet の説明を参照してください。

エラスティック仮想スイッチとゾーンについて

evs および vport プロパティを設定してエラスティック仮想スイッチ (EVS) に接続する anet リソースでは、その anet リソースのプロパティは、evs と vport のペ

ア内にカプセル化されます。EVS anet リソースの次のプロパティは変更できません。

- mac-address
- mtu
- maxbw
- priority
- allowed-address
- vlan-id
- defrouter
- lower-link

EVS anet リソースに設定できる唯一のプロパティは次のとおりです。

- linkname
- evs
- vport
- configure-allowed-address

tenant リソースを設定する必要もあります。テナントは、名前空間の管理に使用されます。tenant 内で定義されている EVS リソースは、そのテナントの名前空間の外部からは見えません。

evszone という名前のゾーンの次の入力、tenantA という名前のテナントの tenant リソースを設定します。zonecfg anet リソースプロパティは、evsa という名前の EVS および vport0 という名前の VPort に接続する anet リソースを持つゾーンの VNIC を作成します。

```
zonecfg:evszone> set tenant=tenantA
```

```
zonecfg:evszone> add anet
```

```
zonecfg:evszone> set evs=EVSA
```

```
zonecfg:evszone> set vport=vport0
```

詳細は、『[Oracle Solaris 11.3 での仮想ネットワークとネットワークリソースの管理](#)』の第 5 章、「[エラスティック仮想スイッチについて](#)」を参照してください。

共有 IP 非大域ゾーン

共有 IP ゾーンは、大域ゾーンからの既存の IP インタフェースを使用します。ゾーンには、1 つ以上の専用の IP アドレスが存在する必要があります。共有 IP ゾーンは、IP

層の構成と状態を大域ゾーンと共有します。次の両方の条件が満たされる場合、ゾーンは共有 IP インスタンスを使用すべきです。

- 大域ゾーンと非大域ゾーンが同じサブネット上に存在するかどうかに関係なく、非大域ゾーンが大域ゾーンによって使用されるデータリンクと同じデータリンクを使用する計画である。
- 排他的 IP ゾーンによって提供されるその他の機能は必要でない。

共有 IP ゾーンには、`zonecfg` コマンドの `net` リソースを使用して、1 つ以上の IP アドレスを割り当てます。大域ゾーンでデータリンク名も構成する必要があります。

`zonecfg net` リソースで、`address` と `physical` のプロパティを設定する必要があります。`defrouter` プロパティはオプションです。

大域ゾーン内で共有 IP タイプのネットワーク接続構成を使用するには、自動ネットワーク構成ではなく `ipadm` を使用する必要があります。ネットワーク接続構成を `ipadm` によって行うかどうかを決定するには、次のコマンドを実行します。表示される応答は、`DefaultFixed` でなければなりません。

```
# svcprop -p netcfg/active_ncp svc:/network/physical:default
DefaultFixed
```

共有 IP ゾーンに割り当てられる IP アドレスは、論理ネットワークインタフェースに関連付けられます。

大域ゾーンから `ipadm` コマンドを使用すると、稼働中のゾーンの論理インタフェースの割り当てまたは削除を行うことができます。

インタフェースを追加するには、次のコマンドを使用します。

```
global# ipadm set-addrprop -p zone=my-zone net0/addr1
```

インタフェースを削除するには、次のコマンドのいずれかを使用します。

```
global# ipadm set-addrprop -p zone=global net0/addr
```

または

```
global# ipadm reset-addrprop -p zone net0/addr1
```

詳細は、『[Oracle Solaris ゾーンの実行と使用](#)』の「共有 IP ネットワークインタフェース」を参照してください。

排他的 IP 非大域ゾーン

排他的 IP は、非大域ゾーンのデフォルトのネットワーク接続構成です。

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。

排他的 IP ゾーン内では、次の各機能を使用できます。

- DHCPv4 および IPv6 ステートレスアドレスの自動構成
- IP フィルタ。ネットワークアドレス変換 (NAT) 機能も含む
- IP ネットワークマルチパス (IPMP)
- IP ルーティング
- TCP/UDP/SCTP および IP/ARP レベルのチューニング可能パラメータを設定するための `ipadm`
- IP セキュリティー (IPsec) と Internet Key Exchange (IKE)。これは、IPsec セキュリティーアソシエーション用の認証済み鍵材料のプロビジョニングを自動化する

排他的 IP ゾーンを構成する方法には、次の 2 つがあります。

- `zonecfg` ユーティリティーの `anet` リソースを使用して、ゾーンのブート時にゾーンに対して一時的な VNIC を自動的に作成し、またゾーンの停止時にそれを削除します。
- 大域ゾーン内にデータリンクを事前構成し、排他的 IP ゾーンにそれを割り当てます。これは、`zonecfg` ユーティリティーの `net` リソースを使用して行います。データリンクは、`net` リソースの `physical` プロパティを使用して指定します。`physical` プロパティは VNIC でもかまいません。`net` リソースの `address` プロパティは設定されません。

データリンクを割り当てると `snoop` コマンドが使用可能になります。

排他的 IP ゾーンは、デフォルトでは関連付けられたインタフェース上のすべての IP アドレスを構成および使用できます。必要に応じて、`allowed-address` プロパティを使用して IP アドレスのコンマ区切りリストを指定することもできます。排他的 IP ゾーンは、`allowed-address` リスト内にはない IP アドレスを使用することはできません。また、ゾーンのブート時に、`allowed-address` リスト内のすべてのアドレスが排他的 IP ゾーンに対して自動で持続的に構成されます。このインタフェース構成が不要な場合、`configure-allowed-address` プロパティを `false` に設定する必要があります。デフォルト値は `true` です。

あるアドレスは自動的に構成する必要があり、またあるアドレスはゾーン内でオンラインおよびオフラインにできる必要がある場合、複数の `anet` リソースを使用できます。たとえば、次の構成には、2 つの `anet` リソースがあります。最初の `anet` リソースは、ゾーンのいずれかのインタフェースでアドレス `192.168.3.3` を自動的に構成します。2 番目の `anet` リソースでは、ゾーンは別のインタフェース上で `192.168.3.100` および `192.168.3.101` のみを構成できます。

```
zonecfg:t> select anet linkname=net0
zonecfg:t:anet> set allowed-address=192.168.3.3/24
zonecfg:t:anet> set configure-allowed-address=true
zonecfg:t:anet> end
zonecfg:t> add anet
zonecfg:t:anet> set allowed-address=192.168.3.100/24,192.168.3.101/24
zonecfg:t:anet> set configure-allowed-address=false
zonecfg:t:anet> end
zonecfg:t>
```

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。dladm コマンドを set-linkprop サブコマンドとともに使用すると、実行中のゾーンに対して追加のデータリンクを割り当てることができます。使用例については、『Oracle Solaris ゾーンの実成と使用』の「排他的 IP 非大域ゾーンでのデータリンクの管理」を参照してください。

独自のデータリンクセットが割り当てられた、実行中の排他的 IP ゾーン内では、ipadm コマンドを使用して IP を構成できます。このコマンドには、論理インタフェースの追加または削除を行う機能が備わっています。ゾーンの IP 構成は、大域ゾーン内の場合と同様に sysconfig インタフェース (sysconfig(1M)) のマニュアルページに記載) を使用して設定できます。

排他的 IP ゾーンの IP 構成は、大域ゾーンから zlogin コマンドを使用することによってのみ表示できます。

```
global# zlogin zone1 ipadm show-addr
ADDROBJ          TYPE      STATE      ADDR
lo0/v4            static    ok          127.0.0.1/8
nge0/v4           dhcp      ok          10.134.62.47/24
lo0/v6            static    ok          ::1/128
nge0/_a           addrconf ok          fe80::2e0:81ff:fe5d:c630/10
```

非大域ゾーンでの Reliable Datagram Sockets のサポート

RDS (Reliable Datagram Sockets) IPC プロトコルは、排他的 IP 非大域ゾーンと共有 IP 非大域ゾーンの両方でサポートされています。RDSv3 ドライバは、SMF サービス rds として有効になっています。デフォルトでは、このサービスはインストール後に無効になります。このサービスは、適切な承認を付与されたゾーン管理者が特定の非大域ゾーン内で有効にすることができます。zlogin のあとで、rds をそれが実行される各ゾーン内で有効にできます。

例 4 非大域ゾーン内で rds サービスを有効にする方法

1. 排他的 IP ゾーンまたは共有 IP ゾーン内で RDSv3 サービスを有効にするには、zlogin コマンドでゾーンにログインし、svcadm enable コマンドを実行します。

```
# svcadm enable rds
```

2. rds が有効になっていることを確認します。

```
# svcs rds
STATE      STIME      FMRI
online     22:50:53  svc:/system/rds:default
```

詳細は、svcadm(1M) のマニュアルページを参照してください。

共有 IP 非大域ゾーンと排他的 IP 非大域ゾーンのセキュリティの相違

共有 IP ゾーン内のアプリケーションは、スーパーユーザーも含め、`zonecfg` ユーティリティーを介してゾーンに割り当てられた IP アドレス以外をソース IP アドレスとしてパケットを送信することはできません。このタイプのゾーンには、任意のデータリンク (レイヤー 2) パケットを送受信するアクセス権はありません。

一方、排他的 IP ゾーンの場合は、`zonecfg` によって指定されたデータリンク全体がゾーンに対して許可されます。結果として、排他的 IP ゾーン内では、`root` ユーザーまたは必要な権利プロファイルを持つユーザーは、大域ゾーン内と同様に、それらのデータリンク上でなりすましのパケットを送信できます。IP アドレスのなりすましは、`allowed-address` プロパティーを設定することによって使用不可にできます。`anet` リソースの場合、`link-protection` プロパティーを設定することによって、`mac-nospoof` や `dhcp-nospoof` などの追加の保護を有効にできます。

共有 IP 非大域ゾーンと排他的 IP 非大域ゾーンの同時使用

共有 IP ゾーンは常に IP 層を大域ゾーンと共有し、排他的 IP ゾーンは常に独自の IP 層インスタンスを持っています。共有 IP ゾーンと排他的 IP ゾーンの両方を同じシステムで使用できます。

ゾーンでマウントされるファイルシステム

各ゾーンには、デフォルトでそのゾーンに委任された ZFS データセットが保持されています。このデフォルトの委任されたデータセットは、デフォルトの大域ゾーンのデータセットレイアウトのデータセットレイアウトを模倣します。`.../rpool/ROOT` というデータセットは、ブート環境を含んでいます。このデータセットは、直接操作しないでください。存在する必要がある `rpool` データセットは、デフォルトで `.../rpool` にマウントされます。`.../rpool/export`、および `.../rpool/export/home` データセットは `/export` と `/export/home` にマウントされます。これらの非大域ゾーンのデータセットは、対応する大域ゾーンのデータセットと使用方法が同じで、同様に管理できます。ゾーン管理者は、`.../rpool`、`.../rpool/export`、および `.../rpool/export/home` のデータセット内に追加のデータセットを作成できます。

`zfs(1M)` のマニュアルページで説明されている `zfs` コマンドを使用して、ゾーンの `rpool/ROOT` ファイルシステムから始まる階層内でファイルシステムの作成、削除、または名前変更を行うべきではありません。`zfs` コマンドは、`canmount`、`mountpoint`、`sharesmb`、`zoned`、`com.oracle.*.*`、`com.sun.*`、および `org.opensolaris.*.*` 以外のプロパティーを設定するために使用できます。

通常、ゾーンでマウントされるファイルシステムには、次のものが含まれます。

- 仮想プラットフォームの初期化時にマウントされる一連のファイルシステム
- アプリケーション環境自体の内部からマウントされる一連のファイルシステム

これらのセットには、たとえば次のようなファイルシステムが含まれます。

- none または legacy 以外の mountpoint があり、また canmount プロパティに yes の値が保持された ZFS ファイルシステム。
- ゾーンの /etc/vfstab ファイルで指定されたファイルシステム。
- AutoFS によるマウントおよび AutoFS によってトリガーされるマウント。autofs プロパティは、sharectl ([sharectl\(1M\)](#)) に記載) を使用して設定します。
- ゾーン管理者が明示的に実行するマウント

また、稼働中のネイティブゾーン内でのファイルシステムのマウント権限は、zonecfg fs-allowed プロパティで定義します。このプロパティは、zonecfg add fs リソースまたは add dataset リソースを使用してゾーンにマウントされたファイルシステムには適用されません。デフォルトでは、ゾーンのデフォルトの委任されたデータセットである hsfs ファイルシステム内でのファイルシステム、および NFS などのネットワークファイルシステムのマウントだけが、ゾーン内で許可されます。



注意 - アプリケーション環境内部から実行される、デフォルト以外のマウントには、いくつかの制限事項があります。これらの制限事項は、ほかのゾーンに悪影響を与えないようにするために、ゾーン管理者がシステムのほかの部分に対するサービスを拒否できないようにします。

一部のファイルシステムについては、ゾーン内部からマウントする場合にセキュリティ制限があります。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。詳細は、『[Oracle Solaris ゾーンの作成と使用](#)』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。

データセットの詳細は、[datasets\(5\)](#) のマニュアルページを参照してください。BE の詳細は、『[Oracle Solaris 11.3 ブート環境の作成と管理](#)』を参照してください。

ファイルシステムのマウントと更新

[pkg\(5\)](#) のマニュアルページで説明されているように、ゾーンのシステムイメージの一部であるファイル、シンボリックリンク、またはディレクトリを非表示にしてファイルシステムをマウントする方法はサポートされていません。/usr/local に内容を提供するパッケージがインストールされていない場合は、/usr/local にファイルシステムをマウントすることが許容されます。しかし、いずれかのパッケージ (SVR4 パッケージを含む) が /usr/local で始まるパスにファイル、ディレクトリ、またはシンボリックリンクを提供する場合は、/usr/local へのファイルシステムのマウントはサ

ポートされません。/mnt へのファイルシステムの一時的なマウントはサポートされています。

ゾーン内でファイルシステムがマウントされる順序のため、/export がゾーンの rpool/export データセットまたは別の委任されたデータセットに由来する場合は、fs リソースが /export/filesys にファイルシステムをマウントすることはできません。

ゾーン内のホスト ID

非大域ゾーンでは、大域ゾーンの hostid とは異なる hostid プロパティを設定できます。これはたとえば、別のシステム上のゾーンに移行された物理マシンなどの場合に行います。このゾーン内に現在あるアプリケーションは、元の hostid に依存している場合があります。詳細は、[46 ページの「リソースタイプとプロパティ」](#)を参照してください。

非大域ゾーンの /dev ファイルシステム

zonecfg コマンドは、規則照合方式を使って、特定のゾーンにどのデバイスを配置するかを指定します。いずれかのルールに一致するデバイスは、ゾーンの /dev ファイルシステムに含まれます。詳細は、『[Oracle Solaris ゾーンの作成と使用](#)』の「[ゾーンの構成方法](#)」を参照してください。

非大域ゾーンのリムーバブル lofi デバイス

リムーバブルループバックファイル (lofi) デバイスは、CD-ROM デバイスのように機能し、非大域ゾーンに構成できます。デバイスがマップするファイルを変更したり、複数の lofi デバイスを作成して、同じファイルを読み取り専用モードで使用したりできます。このタイプの lofi デバイスを作成するには、lofiadm コマンドを -r オプションとともに使用します。作成時にファイル名は必要ありません。リムーバブル lofi デバイスのライフサイクル中は、ファイルを空のデバイスに関連付けたり、空でないデバイスからファイルの関連付けを解除したりできます。1つのファイルを同時に複数のリムーバブル lofi デバイスに安全に関連付けることができます。リムーバブル lofi デバイスは読み取り専用です。通常の読み取り/書き込み lofi デバイスまたはリムーバブル lofi デバイスにマップされたファイルを再マッピングすることはできません。潜在的な lofi デバイスの数は、大域ゾーンで zonecfg コマンドを使用して設定できる zone.max-lofi リソース制御によって制限されます。

作成後のリムーバブル lofi デバイスは読み取り専用です。リムーバブル lofi デバイスに対して書き込み操作が行われると、lofi ドライバはエラーを返します。

lofiadm コマンドは、リムーバブル lofi デバイスを一覧表示する場合にも使用します。

例 5 関連するファイルを含むリムーバブル lofi デバイスを作成する

```
# lofiadm -r /path/to/file
/dev/lofi/1
```

例 6 空のリムーバブル lofi デバイスを作成する

```
# lofiadm -r
/dev/lofi/2
```

例 7 リムーバブル lofi デバイ스에 ファイルを挿入する

```
# lofiadm -r /path/to/file /dev/lofi/1
/dev/lofi/1
```

詳細は、[lofiadm\(1M\)](#)、[zonecfg\(1M\)](#)、および [lofi\(7D\)](#) のマニュアルページを参照してください。表3も参照してください。

非大域ゾーン内でのディスク形式のサポート

ディスクのパーティション分割と `uscsi` コマンドの使用は、`zonecfg` ツールを使用すると有効になります。例については、Resource Type Propertiesの [52 ページの「リソースタイプのプロパティ」](#) を参照してください。`uscsi` コマンドの詳細は、[uscsi\(7I\)](#) を参照してください。

- 委任は、`solaris` ゾーンでサポートされるだけです。
- ディスクは、`-D` オプションを指定した `prtconf` コマンドを使用して表示される `sd` ターゲットを使用する必要があります。[prtconf\(1M\)](#) を参照してください。

ストレージ URI を持つカーネルゾーンのデバイスリソース

次のサポートを利用できます。

- ディスクとして使用されるデバイスがサポートされています。このサポートは、物理ディスク全体、SAN での物理または仮想ディスク全体、Oracle Solaris Cluster と組み合わせたデバイス、および ZFS ボリュームを対象とします。
- カーネルゾーンは、`nfs: URI` を介して NFS ベースのストレージオブジェクトもサポートします。

NFS URI は、指定された NFS ファイル上に作成された `lofi` デバイスに基づいてオブジェクトを指定します。NFS ファイルには `user` および `group` から派生した資格証明を使用してアクセスします。`user` および `group` はユーザー名またはユーザー ID として指定できます。`host` は IPv4 アドレスとして、IPv6 アドレスとして、またはホスト名として指定できます。IPv6 アドレスは、角括弧 ([]) で囲む必要があります。

形式:

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

例:

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

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

- カーネルゾーンは、デバイスリソースで `bootpri` および `id` プロパティをサポートします。
 - ゾーンのルートプールの一部になるディスクでは、`bootpri` のみを設定します。ゾーンのルートプールの一部にならないディスクで `bootpri` を設定した場合、ディスク上のデータが破損する可能性があります。ブート可能にする必要があるデバイスにのみ `bootpri` を設定します。
 - `id` は、カーネルゾーンでディスクのインスタンスを制御します。たとえば、`id=5` は、ディスクがゾーン内で `c1d5` になることを意味します。
- ブータブル `solaris-kz` ディスクで作成されるルート `zpool` は、インストール中に大域ゾーンにインポートできます。この時点で、`zpool` コマンドを使用してルート `zpool` を表示できます。詳細は、[zpool\(1M\)](#) を参照してください。

例 8 移植可能なゾーン構成を作成するためのストレージ URI の構成

デバイスリソースは、ゾーン構成をほかのシステムに移植可能にするストレージ URI を構成する場合にも使用できます。

```
# zonecfg -z my-zone
zonecfg:myzone> add device
zonecfg:myzone:device> set storage=nfs://user1:staff@host1/export/file1
zonecfg:myzone:device> set create-size=4g
```

詳細は、[suri\(5\)](#) のマニュアルページを参照してください。

例 9 現在のデバイスリソース構成の表示

デバイスリソースの現在の構成についての情報を表示するには、`info` サブコマンドを使用してください。たとえば、次のように表示されます。

```
# zonecfg -z my-zone info device
device:
  match not specified
  storage: dev:/dev/zvol/dsk/rpool/VARSHARE/zones/myzone/disk0
  id: 0
  bootpri: 0
device:
  match not specified
  storage: nfs://user1:staff@host1/export/file1
  create-size: 4g
```

ゾーンの ID を指定することによって、特定のゾーンの出力を表示できます。

```
# zonecfg -z my-zone info device id=1
device:
  match not specified
  storage: nfs://user1:staff@host1/export/file1
  create-size: 4g
  id: 1
  bootpri not specified
```

構成可能な特権

ゾーンをブートすると、*safe* 特権のデフォルトセットが構成に含められます。これらの特権は、ゾーン内の特権プロセスがシステムのほかの非大域ゾーン内のプロセスや大域ゾーン内のプロセスに影響を及ぼすことを防ぐため、安全と見なされます。zonecfg コマンドを使用して、次の操作を実行できます。

- デフォルトの特権セットに追加します。ただし、この種の変更を行うと、あるゾーン内のプロセスがグローバルリソースを制御できるようになって、ほかのゾーン内のプロセスに影響する場合があります。
- デフォルトの特権セットから削除します。ただし、この種の変更を行うと、実行に必要な特権がないため一部のプロセスが正しく動作しなくなる場合があります。

注記 - わずかですが、この時点でゾーンのデフォルト特権セットから削除できない特権があります。同じように、特権セットに追加できない特権もあります。

詳細は、『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーン内の特権」、『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」、および [privileges\(5\)](#) を参照してください。

リソースプールの関連付け

『Oracle Solaris 11.3 でのリソースの管理』の第 13 章、「リソースプールの作成と管理のタスク」の説明に従ってシステムでリソースプールを構成した場合、ゾーンの構成

時に `pool` プロパティを使用して、リソースプールの 1 つにゾーンを関連付けることができます。

`dedicated-cpu` リソースを使用すると、非大域ゾーンの実行中にシステムのプロセッサの一部をそのゾーン専用割り当てるよう指定できます。`dedicated-cpu` プロパティを使用すると、CPU、コア、およびソケットをゾーンに割り当てるができます。ゾーンの実行中に使用される一時プールが動的に作成されます。`zonecfg` によって指定すると、移行時にプールの設定が伝達されます。Oracle Solaris カーネルゾーンを構成する場合、`virtual-cpu` リソースも参照してください。

`pool` プロパティを使用すると、同じプールを共有する複数のゾーンを構成できます。

注記 - `pool` プロパティによって設定される持続的プールを使用するゾーン構成と、`dedicated-cpu` リソースによって構成される一時プールには、互換性がありません。これら 2 つのプロパティは、どちらか 1 つしか設定できません。

ゾーン規模のリソース制御の設定

大域管理者または適切な承認を持つユーザーは、ゾーン規模の特権付きリソース制御をゾーンに対して設定できます。ゾーン規模のリソース制御は、ゾーン内のすべてのプロセスエンティティによる総リソース消費を制限します。

これらの制限は、大域ゾーンと非大域ゾーンのどちらに対しても、`zonecfg` コマンドを使用して指定します。『Oracle Solaris ゾーン作成と使用』の「ゾーンの構成方法」を参照してください。

ゾーン規模のリソース制御を設定するためのより簡単な推奨される方法は、`cpu-cap` などの `rctl` リソースではなく、`capped-cpu` などのプロパティ名またはリソースを使用することです。

`zone.cpu-cap` リソース制御は、1 つのゾーンで消費可能な CPU リソースの量に対する絶対的な制限を設定します。設定と同様、100 の値は 1 つの CPU の 100% を意味します。125 の値は 125% になります。CPU キャップの使用時は、100% がシステム上の 1 つの CPU の上限となります。

注記 - `capped-cpu` リソースを設定する場合は、単位に小数を使用できます。この値は `zone.cpu-cap` リソース制御と相互に関連していますが、設定値はその 100 分の 1 になります。設定値 1 はリソース制御の設定値 100 に等しくなります。

`zone.cpu-shares` リソース制御は、公平配分スケジューラ (FSS) の CPU 配分の制限をゾーンに対して設定します。CPU 配分は、まずゾーンに対して割り当てられたあとで、`project.cpu-shares` エントリの指定に従って、ゾーン内のプロジェクトに分配

されます。詳細は、『Oracle Solaris ゾーンの実装と使用』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」を参照してください。この制御のグローバルプロパティ名は `cpu-shares` です。

`zone.max-locked-memory` リソース制御は、1つのゾーンで使用できる、ロックされた物理メモリの量に制限を設定します。ゾーン内のプロジェクト間でのロックされたメモリアリソースの割り当ては、`project.max-locked-memory` リソース制御を使用して制御できます。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。

`zone.max-lofi` リソース制御は、1つのゾーンで作成可能な `lofi` デバイスの数を制限します。

1つのゾーンの LWP が多くなりすぎると、ほかのゾーンに影響を与えることがあります。`zone.max-lwps` リソース制御は、これを防ぐことで、リソースの隔離性を向上させます。ゾーン内のプロジェクト間での LWP リソースの割り当ては、`project.max-lwps` リソース制御を使用して制御できます。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。この制御のグローバルプロパティ名は `max-lwps` です。

`zone.max-processes` リソース制御は、1つのゾーン内のあまりに多くの LWP が他のゾーンに影響を与えることを防ぐことによって、リソースの隔離性を高めます。ゾーン内の各プロジェクトに対するプロセステーブルスロットリソースの割り当ては、`project.max-processes` リソース制御 (『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」に記載) を使用して設定できます。この制御のグローバルプロパティ名は `max-processes` です。`zone.max-processes` リソース制御は、`zone.max-lwps` リソース制御を含むこともできます。`zone.max-processes` が設定されていて `zone.max-lwps` が設定されていない場合、`zone.max-lwps` はゾーンのブート時に暗黙的に `zone.max-processes` の値の 10 倍に設定されます。正常なプロセスとゾンビプロセスの両方がプロセステーブルスロットを占有するため、`max-processes` 制御がプロセステーブルを消費しつくすゾンビプロセスから保護します。ゾンビプロセスには本来 LWP がいないため、`max-lwps` でこの可能性から保護することはできません。

`zone.max-msg-ids`、`zone.max-sem-ids`、`zone.max-shm-ids`、および `zone.max-shm-memory` の各リソース制御は、ゾーン内のすべてのプロセスで使用される System V リソースを制限するために使用されます。ゾーン内のプロジェクト間での System V リソースの割り当ては、これらのリソース制御の `project` バージョンを使用して制御できます。これらの制御のグローバルプロパティ名は、`max-msg-ids`、`max-sem-ids`、`max-shm-ids`、および `max-shm-memory` です。

`zone.max-swap` リソース制御は、ゾーン内のユーザープロセスのアドレス空間マッピングと `tmpfs` マウントで消費されるスワップを制限します。`prstat -z` の出力は「スワップ」列を表示します。報告されるスワップは、ゾーンのプロセスと `tmpfs` 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 `zone.max-swap` 設定を選択できます。

表 3 ゾーン規模のリソース制御

制御名	グローバルプロパティ名	説明	デフォルトの単位	使用される値
zone.cpu-cap		このゾーンに対する CPU リソース量の絶対的な制限。	数量 (CPU の数)、パーセントで表されます。 注記 - capped-cpu リソースとして設定する場合は、単位に小数を使用できます。	
zone.cpu-shares	cpu-shares	このゾーンに対する公平配分スケジューラ (FSS) の CPU 配分	数量 (配分)	
zone.max-locked-memory		ゾーンで使用できるロックされた物理メモリーの合計量。 priv_proc_lock_memory がゾーンに割り当てられている場合、そのゾーンがすべてのメモリーをロックするのを防ぐため、このリソース制御の設定も検討してください。	サイズ (バイト)	capped-memory の locked プロパティ
zone.max-lofi	max-lofi	1 つのゾーンで作成可能な lofi デバイスの数に対する制限	数量 (lofi デバイスの数)	
zone.max-lwps	max-lwps	このゾーンで同時に使用できる LWP の最大数	数量 (LWP 数)	
zone.max-msg-ids	max-msg-ids	このゾーンに許容されるメッセージキュー ID の最大数	数量 (メッセージキュー ID の数)	
zone.max-processes	max-processes	このゾーンで同時に使用できるプロセステーブルスロットの最大数	数量 (プロセステーブルスロット数)	
zone.max-sem-ids	max-sem-ids	このゾーンに許容されるセマフォ ID の最大数	数量 (セマフォ ID の数)	
zone.max-shm-ids	max-shm-ids	このゾーンに許容される共有メモリー ID の最大数	数量 (共有メモリー ID の数)	

制御名	グローバルプロパティ名	説明	デフォルトの単位	使用される値
zone.max-shm-memory	max-shm-memory	このゾーンに許容される System V 共有メモリーの合計量	サイズ (バイト)	
zone.max-swap		このゾーンのユーザープロセスのアドレス空間マッピングと tmpfs マウントで消費できるスワップの合計量。	サイズ (バイト)	capped-memory の swap プロパティ

prctl コマンドを使用すると、実行中のプロセスに対してこれらの制限を指定できます。例が『Oracle Solaris ゾーンの実行と使用』の「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」に記載されています。prctl コマンドで指定された制限には持続性はありません。システムがリブートされると、制限は無効になります。

ゾーンのコメントの追加

attr リソースの型を使ってゾーンのコメントを追加できます。詳細は、『Oracle Solaris ゾーンの実行と使用』の「ゾーンの構成方法」を参照してください。

zonecfg コマンドの使用について

zonecfg コマンド (zonecfg(1M) のマニュアルページに記載) は、非大域ゾーンを構成するために使用します。

zonecfg コマンドは、大域ゾーンのリソース管理設定を持続的に指定するためにも使用できます。たとえば、このコマンドを使用して、dedicated-cpu リソースを使って専用の CPU を使用するように大域ゾーンを構成できます。

zonecfg コマンドは、対話型モード、コマンド行モード、またはコマンドファイルモードで使用できます。このコマンドを使用して、次の操作を実行できます。

- ゾーン構成を作成または削除 (破棄) します
- 特定の構成にリソースを追加します
- 構成に追加したリソースのプロパティを設定します
- 特定の構成からリソースを削除します
- 構成の照会または確認を行います
- 構成を確定します
- 前の構成に戻します

- ゾーンの名前を変更します
- zonecfg のセッションを終了します

zonecfg のプロンプトは次のような形式です。

```
zonecfg:zonename>
```

ファイルシステムなど、特定のリソースタイプの構成を行うときは、そのリソースタイプもプロンプトに表示されます。

```
zonecfg:zonename:fs>
```

この章で説明されているさまざまな zonecfg コンポーネントの使用法を示す手順を含む詳細は、『[Oracle Solaris ゾーンの実装と使用](#)』の第 1 章、「非大域ゾーンの計画および構成方法」を参照してください。

zonecfg のモード

このユーザーインターフェイスでは、有効範囲という概念が使用されます。有効範囲は、大域またはリソース固有のいずれかです。デフォルトの有効範囲は大域です。

大域有効範囲で `add` サブコマンドまたは `select` サブコマンドを使用すると、特定のリソースが選択されます。すると、有効範囲がそのリソースタイプに変わります。

- `add` サブコマンドの場合、`end`、`cancel` のいずれかのサブコマンドを使用すると、リソースの指定が完了します。
- `select` サブコマンドの場合、`end`、`cancel` のいずれかのサブコマンドを使用すると、リソースの変更が完了します。

すると、有効範囲が大域に戻ります。

`add`、`remove`、`set` などのように、有効範囲によって異なる意味を持つサブコマンドもあります。

zonecfg の対話型モード

対話型モードでは、次のサブコマンドがサポートされます。サブコマンドの意味と使用するオプションの詳細は、`zonecfg(1M)` のマニュアルページを参照してください。破壊的な操作や作業内容の消失を伴うようなサブコマンドの場合、処理を実行する前にユーザーの確認が求められます。`-F` (強制) オプションを使用すると、この確認手順を省略できます。

```
help                一般ヘルプまたは特定のリソースに関するヘルプを表示します。
zonecfg:my-zone:capped-cpu> help
```

create	<p>指定された新しいゾーンに使用するメモリー内構成の構成を開始します。次のような目的に使用されます。</p> <ul style="list-style-type: none">■ Oracle Solaris のデフォルト設定を新しい構成に適用します。この方法がデフォルトです。■ <code>-t <i>template</i></code> オプションを使用して、指定したテンプレートと同一の構成を作成します。ゾーン名がテンプレート名から新しいゾーン名に変更されます。■ <code>-F</code> オプションを使用して、既存の構成を上書きします。■ <code>-b</code> オプションを使用して、なにも設定されていない空の構成を作成します。
export	<p>標準出力または指定された出力ファイルに、コマンドファイルに使用できる形式で構成を出力します。</p>
add	<p>大域有効範囲では、指定されたリソースタイプを構成に追加します。</p> <p>リソース固有の有効範囲では、指定された名前と値を持つプロパティを追加します。</p> <p>詳細は、『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」および <code>zonecfg(1M)</code> のマニュアルページを参照してください。</p>
set	<p>指定されたプロパティ名を、指定されたプロパティ値に設定します。ネイティブおよび <code>solaris10</code> ブランドゾーンで使用される <code>zonpath</code> などの一部のプロパティはグローバルですが、リソース固有のプロパティもあることに注意してください。このコマンドは、大域有効範囲とリソース固有の有効範囲の両方で使用できます。</p>
select	<p>大域有効範囲でのみ使用できます。指定されたタイプのリソースのうち、指定されたプロパティ名とプロパティ値の対の条件に一致するものを、変更対象として選択します。有効範囲がそのリソースタイプに変わります。リソースが一意に識別されるように、プロパティの名前と値の対を十分な数だけ指定する必要があります。</p>
clear	<p>省略可能な設定の値をクリアします。必須の設定はクリアできません。ただし、必須の設定のいくつかは、新しい値を割り当てることによって変更できます。プロパティで <code>clear</code> コマンドを使用すると、値がプロパティのデフォルト値にクリアされます。</p>
remove	<p>大域有効範囲では、指定されたリソースタイプを削除します。リソースタイプが一意に識別されるように、プロパティの名前と</p>

値の対を十分な数だけ指定する必要があります。プロパティの名前と値の対をまったく指定しないと、すべてのインスタンスが削除されます。該当するものが複数ある場合は、`-F` オプションを使用していない限り、確認を求めるメッセージが表示されます。

リソース固有の有効範囲では、指定された名前と値を持つプロパティを現在のリソースから削除します。

end	<p>リソース固有の有効範囲でのみ使用できます。リソースの指定を終了します。</p> <p>次に、<code>zonecfg</code> コマンドは、現在のリソースが正しく指定されているかどうかを確認します。</p> <ul style="list-style-type: none"> リソースが正しく指定されている場合は、そのリソースがメモリー内に保持される構成に追加され、有効範囲が大域に戻ります。 指定が不完全な場合は、必要な作業を示すエラーメッセージが表示されます。
cancel	<p>リソース固有の有効範囲でのみ使用できます。リソースの指定を終了し、有効範囲を大域に戻します。リソースの指定が不完全な場合、そのリソースは保持されません。</p>
delete	<p>指定された構成を破棄します。メモリーと安定した記憶領域の両方から構成を削除します。<code>delete</code> に <code>-F</code> (強制) オプションを使用する必要があります。</p>



注意 - この操作は即時に実行されます。確定手順は行われず、削除されたゾーンを元に戻すことはできません。

info	<p>現在の構成または大域のリソースプロパティ <code>zonpath</code>、<code>autoboot</code>、および <code>pool</code> に関する情報を表示します。リソースタイプが指定されている場合は、そのタイプのリソースについてのみ情報を表示します。リソース固有の有効範囲では、このサブコマンドは、追加または変更しようとしているリソースにのみ適用されます。</p>
verify	<p>現在の構成が正しいかどうかを確認します。各リソースに必須プロパティがすべて指定されていることを確認します。<code>rootzpool</code> リソースグループとそのプロパティの構文を確認します。URI によって指定されているストレージのアクセス可能性は確認されません。</p>
commit	<p>現在の構成をメモリーから安定した記憶領域に確定します。メモリー内の構成を確定するまでは、<code>revert</code> サブコマンドで変更内容を削除できます。<code>zoneadm</code> で構成を使用するには、その構成を確</p>

定する必要があります。zonecfg セッションを完了するときに、この操作の実行が自動的に試みられます。正しい構成のみ確定できるので、確定操作では自動的に確認も行われず。

revert	構成を最後に確定されたときの状態に戻します。
exit	zonecfg のセッションを終了します。exit に -F (強制) オプションを使用できます。 必要な場合は、commit 操作が自動的に試行されます。EOF 文字を使ってセッションを終了することもできることに注意してください。

zonecfg のコマンドファイルモード

コマンドファイルモードでは、ファイルから入力されます。このファイルを生成するには、zonecfg Interactive Modeで説明されている [43 ページの「zonecfg の対話型モード」](#) サブコマンドを使用します。構成を標準出力に出力するか、-f オプションで指定した出力ファイルに出力することができます。

ゾーン構成データ

ゾーン構成データは、「リソース」と「プロパティ」という 2 種類のエンティティから成ります。各リソースは、タイプのほかにも 1 つ以上のプロパティを持つことがあります。プロパティは名前と値から成ります。どのようなプロパティセットを持つかは、リソースタイプによって異なります。

リソースタイプとプロパティ

リソースとプロパティのタイプの説明は、次のとおりです。

zonename	<p>ゾーンの名前。ゾーン名には次のような規則が適用されます。</p> <ul style="list-style-type: none"> ■ 各ゾーンの名前は一意でなければならない。 ■ ゾーン名では大文字と小文字が区別される。 ■ ゾーン名は英数字で始まる必要がある。 <p>名前には、英数字、下線 (_)、ハイフン (-)、およびピリオド (.) を使用できます。</p> <ul style="list-style-type: none"> ■ 名前の長さは 63 文字以内でなければならない。 ■ 名前 global は、大域ゾーン用に予約されています。
----------	--

- SYS で始まる名前は予約されており、使用できません。

zonepath

zonecfg template プロパティーで作成されたゾーンでは、zonepath のデフォルト値は /system/zones/zonename です。

指定されている場合、zonepath プロパティーは、ゾーンがインストールされるパスを示します。各ゾーンには、ルートディレクトリのパスが設定されます。これは、大域ゾーンのルートディレクトリに対する相対パスです。インストール時には、大域ゾーンのディレクトリの可視性が制限されている必要があります。ゾーンのパスの所有者は root で、モードは 700 です。ゾーンのパスが存在しない場合、インストール時に自動的に作成されます。アクセス権が正しくない場合、自動的に修正されます。

非大域ゾーンのルートパスは1つ下のレベルになります。ゾーンのルートディレクトリの所有権とアクセス権は、大域ゾーンのルートディレクトリ (/) と同じになります。ゾーンのディレクトリの所有者は root で、モードは 755 であることが必要です。この階層構造により、大域ゾーンのユーザーでも権限を持っていない場合は、非大域ゾーンのファイルシステムと行き来できなくなります。

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。

パス	説明
/system/zones/my-zone	zonecfg zonepath
/system/zones/my-zone/root	ゾーンのルート

詳細は、『Oracle Solaris ゾーンの実装と使用』の「ファイルシステムの行き来」を参照してください。

zonecfg template プロパティーでは、zonepath のデフォルト値は /system/zones/zonename です。

注記 - zoneadm の move サブコマンドで新しいフルパス zonepath を指定することにより、ゾーンを同じシステム上の別の場所に移動できます。手順については、『Oracle Solaris ゾーンの実装と使用』の「非大域ゾーンの移動」を参照してください。

autoboot

このプロパティーを 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されます。これは、デフォルトでは false に設定されます。ゾーンサービス svc:/system/zones:

default が無効になっている場合、このプロパティの設定に関係なく、ゾーンは自動的にブートしません。svcadm(1M)のマニュアルページに記載されているように、svcadm コマンドを使用してゾーンサービスを有効にできます。

```
global# svcadm enable zones
```

pkg update を実行中のこの設定については、『Oracle Solaris ゾーンの作成と使用』の「ゾーンのパッケージングの概要」を参照してください。

autoshtutdown	大域スコープ。大域ゾーンのクリーンシャットダウン時にこのゾーンに対して実行するアクション。値は、shutdown (クリーンゾーンシャットダウン、デフォルト)、halt、または suspend です。
bootargs	このプロパティは、ゾーンのブート引数を設定するために使用します。reboot、zoneadm boot、または zoneadm reboot コマンドで無効にされた場合を除き、このブート引数が適用されます。Zone Boot Argumentsを参照してください。
limitpriv	<p>このプロパティは、デフォルト以外の特権マスクを指定するために使用されます。『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーン内の特権」を参照してください。</p> <p>特権を追加するには、特権名だけを指定するか、特権名の前に priv_ 付けて指定します。特権を除外するには、名前の前にダッシュ (-) または感嘆符 (!) を付けます。複数の特権は、コンマで区切り、引用符 (") で囲みます。</p> <p>priv_str_to_set(3C) で説明されているように、特殊な特権セット none、all、および basic は、それぞれの通常の定義に展開されます。ゾーン構成は大域ゾーンで行われるため、特殊な特権セット zone は使用できません。特定の特権を追加または削除してデフォルトの特権セットを変更するのが一般的な使用方法であるため、特殊なセットである default はデフォルトの特権セットにマップされます。limitpriv プロパティの先頭に default がある場合、デフォルトセットに展開されます。</p> <p>次のエントリは、dtrace_proc 特権と dtrace_user 特権だけを必要とする DTrace プログラムをゾーンで使用できるようにします。</p> <pre>global# zonecfg -z userzone zonecfg:userzone> set limitpriv="default,dtrace_proc,dtrace_user"</pre> <p>次のエントリでは、priocntl1 コマンドを使用して、システム上のアクティブなプロセス、タスク、またはプロジェクトに関連付けられたリソース制御を調べて変更できます。</p> <pre>global# zonecfg -z userzone</pre>

	<pre>zonecfg:userzone> set limitpriv="default,proc_priocntl"</pre> <p>ゾーンの特権セットに不許可の特権が含まれる場合、必須の特権が欠落している場合、または未知の特権が含まれる場合、ゾーンの検証、準備、またはブートの試行は失敗し、エラーメッセージが表示されます。</p>
scheduling-class	このプロパティは、ゾーンのスケジューリングクラスを設定します。詳細とヒントについては、 19 ページの「スケジューリングクラス」 を参照してください。
ip-type	このプロパティは、すべての非大域ゾーンのために設定する必要があります。 30 ページの「排他的 IP 非大域ゾーン」 、 29 ページの「共有 IP 非大域ゾーン」 、および『 Oracle Solaris ゾーンの作成と使用 』の「 ゾーンの構成方法 」を参照してください。
dedicated-cpu	このリソースは、ゾーンの実行中にシステムのプロセッサの一部をそのゾーン専用に割り当てます。dedicated-cpu リソースは、ncpus の制限を設定し、必要に応じて importance、ncores、cores、および sockets も設定します。詳細は、 16 ページの「dedicated-cpu リソース」 を参照してください。
solaris-kz のみ: virtual-cpu	この solaris-kz リソースは、ゾーンの実行中にシステムのプロセッサの一部をそのゾーン専用に割り当てます。virtual-cpu リソースは、ncpus の制限を設定します。詳細は、 17 ページの「solaris-kz のみ: virtual-cpu リソース」 を参照してください。
capped-cpu	このリソースは、ゾーンの実行中にゾーンで消費可能な CPU リソースの量に対する制限を設定します。この capped-cpu リソースは、ncpus に制限を設けます。詳細は、 18 ページの「capped-cpu リソース」 を参照してください。
capped-memory	このリソースは、ゾーンのメモリー上限を設定する際に使用される各プロパティをグループ化します。capped-memory リソースは、physical、swap、および locked メモリーの制限を設定します。これらのプロパティの少なくとも 1 つは指定する必要があります。capped-memory リソースを使用するには、service/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。
solaris-kz のみ: ib-vhca	ib-vhca リソースは、排他的 IP ゾーンのブート時に、そのゾーンに対して一時的な仮想 InfiniBand HCA デバイスを自動的に作成します。デバイスは、ゾーンが停止すると削除されます。

『Oracle Solaris 11.3 での仮想ネットワークとネットワークリソースの管理』も参照してください。

anet	<p>anet リソースは、排他的 IP ゾーンのブート時に、そのゾーンに対して一時的な VNIC インタフェースを自動的に作成します。VNIC は、ゾーンが停止すると削除されます。</p>
net	<p>net リソースは、非大域ゾーンに大域ゾーン内にある既存のネットワークインタフェースを割り当てます。ネットワークインタフェースリソースは、インタフェースの名前です。各ゾーンでは、インストール済み状態から準備完了状態に移行するときに設定される、複数のネットワークインタフェースを保持できます。</p>
solaris、solaris10 のみ: dataset	<p>dataset リソースで使用する必要のある唯一のデータセットタイプは、ZFS™ ファイルシステムです。ZFS dataset リソースを追加して、ストレージ管理の非大域ゾーンへの委任を有効にします。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。ゾーン管理者は、子の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、ゾーンに追加されていないデータセットを操作したり、ゾーンに割り当てられているデータセットに設定されている最上位レベルの割り当て制限を超過したりすることはできません。データセットが非大域ゾーンに委任されると、zoned プロパティが自動的に設定されます。ゾーン管理者がマウントポイントを受け入れられない値に設定しなければならない可能性があるため、ゾーンファイルシステムを大域ゾーンにマウントすることはできません。</p> <p>次の方法で、ZFS データセットをゾーンに追加できます。</p> <ul style="list-style-type: none"> ■ lofs マウントされたファイルシステムとして (大域ゾーンとの領域共有のみが目的の場合) ■ 委任されたデータセットとして
	<p>zonecfg template プロパティの使用中に rootzpool リソースが指定されていない場合、デフォルトの zonpath データセットは <i>rootpool/VARSHARE/zones/zonename</i> です。データセットは、マウントポイント <i>/system/zones</i> を持つ svc-zones サービスによって作成されます。残りのプロパティは、<i>rootpool/VARSHARE/zones/</i> から継承されます。</p>
	<p>『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」、『Oracle Solaris ゾーンの作成と使用』の「ファイルシステムと非大域ゾーン」、および datasets(5) のマニュアルページを参照してください。</p>

データセットの問題については、『Oracle Solaris ゾーンの作成と使用』の第12章、「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」を参照してください。

注記 - カーネルゾーンでは `dataset` リソースではなく `device` リソースを使用します。

`fs` 各ゾーンでは、インストール済み状態から準備完了状態に移行するときにマウントする各種のファイルシステムを指定できます。ファイルシステムリソースは、ファイルシステムのマウントポイントのパスを指定します。ゾーンでのファイルシステムの使用の詳細は、『Oracle Solaris ゾーンの作成と使用』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。

注記 - 非大域ゾーン内の UFS ファイルシステムを `fs` リソースによって使用するには、インストールの後か、または AI マニフェストスクリプトを使用して、`system/filesystem/ufs` パッケージをゾーンにインストールする必要があります。

`fs` リソースによって追加された UFS ファイルシステムの割り当て制限情報を取得するために、`quota` コマンド (`quota(1M)` を参照) は使用できません。

`solaris` および
`solaris10` のみ:
`fs-allowed`

このプロパティを設定すると、ゾーン管理者によって作成されたか、または NFS を使用してインポートされた、該当のタイプの任意のファイルシステムをマウントし、またそのファイルシステムを管理するための機能がゾーン管理者に提供されます。また、稼働中のゾーン内でのファイルシステムのマウント権限は、`fs-allowed` プロパティで制限します。デフォルトでは、`hsfs` ファイルシステムおよび NFS などのネットワークファイルシステムのマウントだけがゾーン内で許可されます。

このプロパティは、ゾーンに委任されたブロック型デバイスでも使用できます。

`fs-allowed` プロパティは、ゾーン内からマウントできる、追加のファイルシステムのコンマ区切りリスト (たとえば、`ufs`, `pcfs`) を受け入れます。

```
zonecfg:my-zone> set fs-allowed=ufs,pcfs
```

このプロパティは、`add fs` プロパティまたは `add dataset` プロパティを使用して大域ゾーンによって管理されるゾーンのマウントに影響を与えません。

セキュリティに関する考慮事項については、『Oracle Solaris ゾーンの作成と使用』の「ファイルシステムと非大域ゾーン」および『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーンでのデバイスの使用」を参照してください。


```
zonecfg:my-zone> add admin
zonecfg:my-zone:admin> set user=zadmin
zonecfg:my-zone:admin> set auths=login,manage
zonecfg:my-zone:admin> end
```

auths プロパティには、次の値を使用できます。

- login (solaris.zone.login)
- manage (solaris.zone.manage)
- clone (solaris.zone.clonefrom)
- config (solaris.zone.config)
- config (solaris.zone.liveconfig)

これらの auths はゾーンの作成を許可するわけではありません。この機能は、ゾーンのセキュリティープロファイル内に含まれています。

solaris および
solaris10 のみ:
rootzpool

storage

ゾーンのインストール専用の ZFS zpool を提供するためのストレージオブジェクトの URI を識別します。URI と storage に指定可能な値については、22 ページの「[solaris および solaris10 のみ: rootzpool リソース](#)」を参照してください。ゾーンのインストール時に、zpool が自動的に作成されるか、事前に作成された zpool がインポートされます。my-zone_rpool という名前が割り当てられます。

```
zonecfg:my-zone> add rootzpool
zonecfg:my-zone:rootzpool> add storage dev:dsk/c4t1d0
zonecfg:my-zone:rootzpool> end
```

ミラー化構成を作成する場合は、storage プロパティを追加できます。

```
add storage dev:dsk/c4t1d0
add storage dev:dsk/c4t3d0
```

1 つのゾーンに構成できる rootzpool リソースは 1 つだけです。

solaris および
solaris10 のみ:
zpool

storage、name

zpool をゾーンに委任するためのストレージオブジェクトの URI を 1 つ以上定義します。URI と storage プロパティに指定可能な値については、22 ページの「[solaris および solaris10 のみ: rootzpool リソース](#)」を参照してください。name プロパティに指定可能な値は、zpool(1M) のマニュアルページで定義されています。

この例では、zpool ストレージリソースがゾーンに委任されます。zpool は自動的に作成されるか、または以前に作成された zpool がインストール中にインポートされます。zpool の名前は my-zone_pool1 です。

```
zonecfg:my-zone> add zpool
zonecfg:my-zone:zpool> set name=pool1
zonecfg:my-zone:zpool> add storage dev:dsk/c4t2d0
zonecfg:my-zone:zpool> add storage dev:dsk/c4t4d0
zonecfg:my-zone:zpool> end
```

ゾーン構成には、1つ以上の zpool リソースを含めることができます。

dedicated-cpu

ncpus、importance、cores、cpus、sockets

CPU の数を指定し、必要に応じてプールの相対的な重要性も指定します。次の例では、ゾーン my-zone で使用する CPU の範囲を指定します。importance も設定します。

```
zonecfg:my-zone> add dedicated-cpu
zonecfg:my-zone:dedicated-cpu> set ncpus=1-3
zonecfg:my-zone:dedicated-cpu> set importance=2
zonecfg:my-zone:dedicated-cpu> end
```

コア 0、1、2、および 3 を永続的にゾーン my-zone に割り当てます。次の dedicated-cpu の例では cores を使用しますが、cpus=、cores=、および sockets= はすべて使用可能です。

```
zonecfg:my-zone> add dedicated-cpu
zonecfg:my-zone:dedicated-cpu> set cores=0-3
zonecfg:my-zone:dedicated-cpu> end
```

virtual-cpu

ncpus

CPU の数を指定します。次の例では、ゾーン my-zone に 3 つの CPU を指定します。

```
zonecfg:my-zone> add virtual-cpu
zonecfg:my-zone:dedicated-cpu> set ncpus=3
zonecfg:my-zone:dedicated-cpu> end
```

capped-cpu

ncpus

CPU の数を指定します。次の例では、ゾーン my-zone の CPU 数のキャップを 3.5 に指定します。

```
zonecfg:my-zone> add capped-cpu
zonecfg:my-zone:capped-cpu> set ncpus=3.5
zonecfg:my-zone:capped-cpu> end
```

capped-memory

physical、swap、locked

ゾーン my-zone のメモリー制限を指定します。各制限はオプションですが、少なくとも 1 つは設定する必要があります。

```
zonecfg:my-zone> add capped-memory
zonecfg:my-zone:capped-memory> set physical=50m
zonecfg:my-zone:capped-memory> set swap=100m
zonecfg:my-zone:capped-memory> set locked=30m
zonecfg:my-zone:capped-memory> end
```

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。

fs

dir、special、raw、type、options

fs リソースのパラメータは、ファイルシステムをマウントする方法と場所を決定する値を指定します。fs のパラメータは次のように定義されています。

dir	ファイルシステムのマウントポイントを指定します
special	大域ゾーンからマウントするブロック型特殊デバイスの名前またはディレクトリを指定します
raw	ファイルシステム (ZFS には適用されない) をマウントする前に、fsck の実行対象の raw デバイスを指定します。
type	ファイルシステムのタイプを指定します
options	mount コマンドで使用されるオプションに似たマウントオプションを指定します

次の例の各行では、大域ゾーン内の pool1/fs1 という名前のデータセットを、構成されるゾーン内で /shared/fs1 としてマウントすることを指定しています。使用するファイルシステムのタイプは ZFS です。

```
zonecfg:my-zone> add fs
zonecfg:my-zone:fs> set dir=/shared/fs1
zonecfg:my-zone:fs> set special=pool1/fs1
zonecfg:my-zone:fs> set type=zfs
zonecfg:my-zone:fs> end
```

パラメータの詳細は、『Oracle Solaris ゾーンの作成と使用』の「-o nosuid オプション」、『Oracle Solaris ゾーンの作成と使用』の「セキュリティの制限およびファイルシステムの動作」、および fsck(1M) と mount(1M) のマニュアルページを参照してください。また、セクション 1M のマニュアルページには、特定のファイルシステムに固有のマウントオプションに関するものがあります。このようなマニュアルページの名前は、mount_filesystem という形式です。

注記 - この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制限情報を取得するために、`quota` コマンド ([quota\(1M\)](#) を参照) は使用できません。

dataset

name、alias

次の例の各行では、データセット `sales` を非大域ゾーンでマウントして可視にし、大域ゾーンでは不可視にするように指定しています。

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

委任されたデータセットは、次の例に示されているようにデフォルト以外の別名を持つことができます。データセットの別名には、スラッシュ (/) を含めることができないことに注意してください。

```
zonecfg:my-zone> add dataset
zonecfg:my-zone:dataset> set name=tank/sales
zonecfg:my-zone:dataset> set alias=data
zonecfg:my-zone:dataset> end
```

`%{zonename}` トークンは、`name` プロパティに使用できます。デフォルトの別名に戻すには、`clear alias` を使用します。

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

solaris-kz のみ:
ib-vhca

over-hca、id、port

`ib-vhca` リソースは、仮想機能 (VF) を割り当てるために使用する物理機能 (PF) を指定します。

カーネルゾーンで VF を割り当てるには、次の手順を使用します。

1. `ibadm(1M)` のマニュアルページに記載の `ibadm` コマンドを使用して、PF を仮想化します。
2. `zonecfg` コマンドを使用して、カーネルゾーンに VF を割り当てます。特定の VF インデックスが指定されていないことに注意してください。ブート時に、使用可能な VF が、`zoneadmd` によって指定された PF からカーネルゾーンに動的に割り当てられます。VF が使用可能ではない場合、リソースの割り当ては失敗します。

over-hca

仮想 InfiniBand デバイスの構成のために使用する物理 InfiniBand デバイスを設定します。デバイス名を取得するには、`ibadm(1M)` を参照してください。

id	ib-vhca リソースの一意的識別子。
port	<p>ポートリソースを使用して、割り当てられた VF に許容可能な pkey を指定します。ポートには、物理ポート番号 (通常 1 または 2) に対応する id プロパティもあります。</p> <p>pkey InfiniBand パーティションキーを指定します。pkey 値は、キーワードか、16 進数値のコンマ区切りリストのいずれかになります。16 進値を指定するときに、0x の接頭辞は使用しないでください。</p> <p>pkey に使用されるキーワードは auto です。auto キーワードを使用して、指定された over-hca の値に基づいて pkey 値を自動的に生成して割り当てます。</p> <p>id id 値は、port リソースを一意的に識別するために使用されます。id は物理ポート番号に対応します。</p>
anet	<p>linkname、lower-link、allowed-address、auto-mac-address、configure-allowed-address、defrouter linkmode (IPoIB)、mac-address (IPoIB 以外)、mac-slot (IPoIB 以外)、mac-prefix (IPoIB 以外)、mtu、maxbw、pkey (IPoIB)、priority、vlan-id (IPoIB 以外)、rxfanout、rxrings、txrings、link-protection、allowed-dhcp-cids</p> <p>solaris-kz のみ: zonecfg anet リソースの iov プロパティを使用すると、シングルルート I/O (SR-IOV) NIC 仮想機能 (VF) を作成および管理できます。次のいずれかのプロパティが設定されている場合は、iov プロパティを auto または on に設定しないでください。</p> <ul style="list-style-type: none"> ■ allowed-address ■ allowed-dhcp-cids ■ configure-allowed-address ■ cos ■ defrouter ■ etsbw-lcl ■ evs ■ link-protection ■ maxbw ■ mtu

- priority
- rxfanout
- rxrings
- txrings
- vlan-id
- vport
- vsi-mgrid
- vsi-typeid
- vsi-vers

iov プロパティがすでに auto または on に設定されている場合は、これらのプロパティの設定に失敗します。

例と詳細については、『[Oracle Solaris カーネルゾーンの作成と使用](#)』の「[カーネルゾーンでのシングルルート I/O NIC 仮想化の管理](#)」および [zonecfg\(1M\)](#) のマニュアルページを参照してください。

注記 - カーネルゾーンのウォーム移行では、`zonecfg iov` プロパティが auto または on に設定されている場合、中断および再開の操作はサポートされません。カーネルゾーンでの中断および再開の操作の詳細は、『[Oracle Solaris カーネルゾーンの作成と使用](#)』の「[suspend リソースの構成](#)」および『[Oracle Solaris カーネルゾーンの作成と使用](#)』の「[ウォーム移行を使用したカーネルゾーンの移行](#)」を参照してください。

solaris のみ: `zonecfg` では、IPoIB データリンクに次の `anet` プロパティを設定しないでください。

- mac-address
- mac-prefix
- mac-slot
- vlan-id

`zonecfg` では、IPoIB 以外のデータリンクに次の `anet` プロパティを設定しないでください。

- linkmode
- pkey

EVS `anet` リソースには、次のプロパティのみを設定してください。

- linkname
- evs
- vport
- configure-allowed-address

anet リソースは、ゾーンのブート時に自動 VNIC インタフェースまたは IPoIB インタフェースを作成し、ゾーンの停止時にその VNIC または IPoIB インタフェースを削除します。solaris-kz ブランドでは IPoIB はサポートされません。リソースプロパティは、zonecfg コマンドを使用して管理されます。使用できるプロパティのすべてのテキストについては、[zonecfg\(1M\)](#) のマニュアルページを参照してください。

iov	<p>『Oracle Solaris カーネルゾーンの作成と使用』の「カーネルゾーンでのシングルルート I/O NIC 仮想化の管理」を参照してください。ネットワーク統計情報を提供するために使用されるシャドウ VNIC に関する具体的な情報については、『Oracle Solaris カーネルゾーンの作成と使用』の「Oracle Solaris カーネルゾーンでの仮想機能とシャドウ VNIC の使用」を参照してください。</p>
lower-link	<p>作成されるリンクの基礎となるリンクを指定します。auto に設定すると、zoneadmd デモンは、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。anet リソースの lower-link として VNIC を作成できる任意のリンクを指定できます。</p> <p>ブート中に VNIC を自動的に作成するためのデータリンクを選択すると、すべての IPoIB リンクがスキップされます。</p>
linkname	<p>自動作成される VNIC インタフェースまたは IPoIB インタフェースの名前を指定します。solaris-kz では IPoIB はサポートされません。</p>
mac-address (IPoIB には 非適用)	<p>指定された値またはキーワードに基づいて VNIC の MAC アドレスを設定します。値がキーワードでない場合は、ユニキャスト MAC アドレスとして解釈されます。サポートされるキーワードについては、zonecfg(1M) のマニュアルページを参照してください。ランダムな MAC アドレスを選択すると、生成されたアドレスはゾーンの再ブートおよびゾーンの切り離しと接続の操作後にも保持されます。デフォルトのポ</p>

リシー `auto-mac-address` を使用するとき
は、Oracle Solaris ゾーンはランダムな `mac-`
`address` を取得できます。

pkey (IPoIB のみ)	IPoIB データリンクインタフェースを作成するために使用するパーティションキーを設定します。このプロパティは必須です。指定した pkey は、0x 接頭辞があるかどうかに関係なく、常に 16 進数として扱われます。
linkmode (IPoIB のみ)	データリンクインタフェースの linkmode を設定します。デフォルト値は cm です。有効な値は次のとおりです。
cm (デフォルト)	接続モード。このモードでは、65520 バイトのデフォルトの MTU を使用し、65535 バイトの最大 MTU をサポートします。
ud	低信頼データグラムモード。リモートノードに対して接続モードが使用できない場合は、代わりに低信頼データグラムモードが自動的に使用されます。このモードでは、2044 のデフォルトの MTU を使用し、4092 バイトの最大 MTU をサポートします。
allowed-address	排他的 IP ゾーン用の IP アドレスを構成し、また排他的 IP ゾーンで使用できる、構成可能な IP アドレスセットの制限も行います。複数のアドレスを指定するには、IP アドレスのコンマ区切りリストを使用します。
defrouter	defrouter プロパティを使用すると、非大域ゾーンと大域ゾーンが別個のネットワーク上に存在する場合にデフォルト経路を設定できます。

defrouter プロパティが設定されたゾーンは、大域ゾーン用に構成されていないサブネット上に存在する必要があります。

zonecfg コマンドで SYSdefault テンプレートを使用してゾーンを作成したときに、ほかの IP リソースが設定されていない場合は、次のプロパティを持つ anet リソースがゾーン構成に自動的に組み込まれます。linkname は、物理 Ethernet リンクに対して自動的に作成され、netN という形式の最初に使用できる名前 (net0) に設定されます。デフォルト値を変更するには、zonecfg コマンドを使用します。

デフォルトのポリシー auto を使用すると、適切な mac-address が割り当てられます。

Oracle Solaris ゾーン ランダムな mac-address

Oracle Solaris カーネルゾーン ランダムな mac-address

カーネルゾーンの下
の Oracle Solaris ゾーン ファクトリ mac-address

Oracle VM Server for SPARC ゲストドメイン ファクトリ mac-address

Oracle VM Server for SPARC ゲストドメインで実行されている Oracle Solaris カーネルゾーン ファクトリ mac-address

デフォルトのポリシーでは、物理 Ethernet リンク (たとえば、net0) に対して自動 VNIC が作成され、この VNIC に MAC アドレスが割り当てられます。オプションの lower-link プロパティは、自動 VNIC が作成される、基礎となるリンク vnic1 に設定されます。zonecfg コマンドを使用して、リンク名、ベースとなる物理リンク、MAC アドレス、帯域幅制限などの VNIC プロパティ、およびその他の VNIC プロパティを指定できます。ip-type=exclusive も指定する必要があります。

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone> add anet
zonecfg:my-zone:anet> set linkname=net0
zonecfg:my-zone:anet> set lower-link=auto
zonecfg:my-zone:anet> set mac-address=random
zonecfg:my-zone:anet> set link-protection=mac-nospoof
zonecfg:my-zone:anet> end
```


共有 IP ゾーンの場合の次の例では、物理インタフェース nge0 が、IP アドレス 192.168.0.1 のゾーンに追加されています。システム上のネットワークインタフェースをリスト表示するには、次のように入力します。

```
global# ipadm show-if -po ifname,class,active,persistent
lo0:loopback:yes:46--
nge0:ip:yes:----
```

出力の各行には、ループバックの行を除いて、ネットワークインタフェースの名前があります。説明に loopback が含まれている行は、カードには当てはまりません。46 の persistent フラグは、そのインタフェースが大域ゾーン内で持続的に構成されていることを示します。yes という有効な値は、そのインタフェースが現在構成されていることを示しています。また class の ip の値は、nge0 がループバックインタフェースではないことを示しています。ゾーンのデフォルト経路は、10.0.0.1 に設定されています。defrouter プロパティの設定はオプションです。ip-type=shared は必須です。

```
zonecfg:my-zone> set ip-type=shared
zonecfg:my-zone> add net
zonecfg:my-zone:net> set physical=vnic1
zonecfg:my-zone:net> set address=192.168.0.1
zonecfg:my-zone:net> set defrouter=10.0.0.1
zonecfg:my-zone:net> end
```

排他的 IP ゾーンの場合の次の例では、VNIC が、VLAN である物理インタフェースに使用されます。使用可能なデータリンクを調べるには、dladm show-link を使用してください。allowed-address プロパティは、ゾーンが使用できる IP アドレスを抑制します。defrouter プロパティは、デフォルト経路を設定するために使用します。ip-type=exclusive も指定する必要があります。

```
zonecfg:my-zone> set ip-type=exclusive
zonecfg:my-zone> add net
zonecfg:myzone:net> set allowed-address=10.1.1.32/24
zonecfg:my-zone:net> set physical=vnic1
zonecfg:myzone:net> set defrouter=10.1.1.1
zonecfg:my-zone:net> end
```

add net 手順では、物理デバイスタイプだけを指定します。physical プロパティは VNIC でもかまいません。

注記 - Oracle Solaris オペレーティングシステムは、Ethernet タイプのすべてのインタフェースをサポートしています。dladm コマンドを使用してデータリンクを管理できます。

device match、allow-partition、allow-raw-io

マッチングするデバイス名は、マッチングするパターンまたは絶対パスにできます。次のトークンは、`match` および `storage` プロパティーでサポートされています。

- `%{zonename}`
- `%{id}`
- `%{ global-rootzpool}`

`allow-partition` と `allow-raw-io` の両方を `true` または `false` に設定できます。デフォルトは `false` です。`allow-partition` はパーティション分割を有効にします。`allow-raw-io` は `uscsi` を有効にします。

これらのリソースの詳細は、[zonecfg\(1M\)](#) を参照してください。

`solaris-kz` ゾーンについて `device:match` リソースプロパティーで指定できる制限には、次のものが含まれます。

- LUN ごとに1つのリソースのみが許可されます。
- スライスとパーティションはサポートされません。
- `raw` ディスクデバイスのサポートのみが提供されます。
- サポートされるデバイスパスは、`lofi`、`ramdisk`、`dsk`、および `zvol` です。

次の例では、ディスクデバイスに対する `uscsi` 操作が `solaris` ゾーン構成に追加されます。

```
zonecfg:my-zone> add device
zonecfg:my-zone:device> set match=/dev/*dsk/cXtYdZ*
zonecfg:my-zone:device> set allow-raw-io=true
zonecfg:my-zone:device> end
```

Veritas Volume Manager デバイスは、`add device` を使用して非大域ゾーンに委任されます。

次の例では、ストレージデバイスが `solaris-kz` ゾーンに追加されています。

```
zonecfg:my-zone> add device
zonecfg:my-zone:device> set storage=iscsi:///lunname.naa.
600144f03d70c80000004ea57da10001
zonecfg:my-zone:device> set bootpri=0
zonecfg:my-zone:device> end
```

`storage` プロパティーのトークンを使用している場合は、`device` リソースの新しいインスタンスがゾーン構成に追加されると、システムには次のように表示されます。

```
device 0:
  match not specified
  storage.template: dev:/dev/zvol/dsk/{global-rootzpool}/
VARSHARE/zones/{zonename}/disk{id}
```

```

storage: dev:/dev/zvol/dsk/rpool/VARSHARE/zones/kernel-zone1/
disk0
  id: 0
  bootpri: 0

```

storage は、デフォルト値を持つ唯一のプロパティであるため、このプロパティにのみ、リソースの追加後に表示される info 出力の値が含まれています。



注意 - デバイスを追加する前に、制限とセキュリティーに関する確認事項について、『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーンでのデバイスの使用」、『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーンでのアプリケーションの実行」、および『Oracle Solaris ゾーンの作成と使用』の「非大域ゾーン内の特権」を参照してください。

rctl name、value
使用可能なゾーン規模のリソース制御は、次のとおりです。

- zone.cpu-cap
- zone.cpu-shares (推奨: cpu-shares)
- zone.max-locked-memory
- zone.max-lofi
- zone.max-lwps (推奨: max-lwps)
- zone.max-msg-ids (推奨: max-msg-ids)
- zone.max-processes (推奨: max-processes)
- zone.max-sem-ids (推奨: max-sem-ids)
- zone.max-shm-ids (推奨: max-shm-ids)
- zone.max-shm-memory (推奨: max-shm-memory)
- zone.max-swap

ゾーン規模のリソース制御を設定するために推奨される、より簡単な方法は、『Oracle Solaris ゾーンの作成と使用』の「ゾーンの構成方法」に示されているように、rctl リソースの代わりにプロパティ名を使用する方法です。add rctl を使用してゾーン内のゾーン規模のリソース制御エントリを構成する場合、その形式は project データベース内のリソース制御エントリの形式とは異なります。ゾーン構成では、rctl リソースタイプは、名前と値の対 3 つから成ります。これらの名前は、priv、limit、および action です。これらの名前には、単純な値がそれぞれ設定されます。

```

zonecfg:my-zone> add rctl
zonecfg:my-zone:rctl> set name=zone.cpu-shares
zonecfg:my-zone:rctl> add value (priv=privileged,limit=10,action=none)
zonecfg:my-zone:rctl> end

```

```
zonecfg:my-zone> add rctl
zonecfg:my-zone:rctl> set name=zone.max-lwps
zonecfg:my-zone:rctl> add value (priv=privileged,limit=100,action=deny)
zonecfg:my-zone:rctl> end
```

リソース制御と属性の全般的な情報については、『[Oracle Solaris 11.3 のリソースの管理](#)』の第6章、「リソース制御について」および『[Oracle Solaris ゾーンの作成と使用](#)』の「非大域ゾーンで使用されるリソース制御」を参照してください。

attr name、type、value

次の例では、ゾーンに関するコメントを追加します。

```
zonecfg:my-zone> add attr
zonecfg:my-zone:attr> set name=comment
zonecfg:my-zone:attr> set type=string
zonecfg:my-zone:attr> set value="Production zone"
zonecfg:my-zone:attr> end
```

export サブコマンドを使用すると、ゾーン構成を標準出力に出力できます。構成は、コマンドファイルに使用できる形式で保存されます。

Tecla コマンド行編集ライブラリ

付属の Tecla コマンド行編集ライブラリは、**zonecfg** コマンドで使用できます。このライブラリにより、コマンド行の履歴メカニズムと編集サポートが提供されます。

詳細については、[tecla\(5\)](#) のマニュアルページを参照してください。

索引

あ

一時プール, 16

か

カーネルゾーン

CPU 間の移行, 25, 25

host-compatible, 25, 26

virtual-cpu リソース, 17

機能

排他的 IP ゾーン, 30

共有 IP ゾーン, 29

構成可能な特権、ゾーン, 38

公平配分スケジューラ (FSS), 19

さ

スワップ領域の上限, 20

ゾーン, 57

anet, 50, 57

autoshtutdown プロパティ, 48

bootargs プロパティ, 48

capped-cpu, 49

capped-memory, 19, 49

dedicated-cpu, 49

ib-vhca, 49, 56

ip-type, 49

limitpriv, 48

net, 50

npiv, 52

pool, 52

rootzpool, 53

scheduling-class, 49

virtual-cpu, 49

共有 IP, 29

権利、役割、プロファイル, 10

構成, 42

構成可能な特権, 38

構成の概要, 13

公平配分スケジューラ (FSS), 19

ゾーン規模のリソース制御, 46

ディスク形式のサポート, 36

データセット, 50

排他的 IP, 30

プロパティタイプ, 46

リソース制御, 39

リソースタイプ, 46

リソースタイプのプロパティ, 52

ゾーン規模のリソース制御, 39

ゾーンの admin 承認, 15

ゾーンを使用した構成, 62

ゾーン capped-cpu, 18

た

ディスク形式のサポート

ゾーン, 36

データリンク, 28

デバイスリソース

ストレージ URI, 36

は

排他的 IP ゾーン, 30

物理メモリーの上限, 20

や

読み取り専用ゾーン

file-mac-profile, 15
読み取り専用ゾーンルート, 15

ら

リソース制御
 ゾーン規模, 39
リムーバブル lofi デバイス, 35
ロックされたメモリーの上限, 20

A

allowed-addresses
 排他的 IP ゾーン, 30
autoboot, 14
autoshtutdown プロパティ, 48

B

bootargs プロパティ, 48

C

capped-cpu リソース, 18, 49
capped-memory, 49
capped-memory リソース, 19
CPU 間の移行
 カーネルゾーン, 25, 25

D

dedicated-cpu リソース, 16, 49
defrouter, 62
 排他的 IP ゾーン, 30
DHCP
 排他的 IP ゾーン, 31
dtrace_proc, 48
dtrace_user, 48

E

EVS

ゾーン, 28

G

global-time, 14

H

host-compatible
 カーネルゾーン, 25, 26
hostid, 35

I

ip-type プロパティ, 49
IP フィルタ
 排他的 IP ゾーン, 31
IP ルーティング
 排他的 IP ゾーン, 31
IPMP
 排他的 IP ゾーン, 31
IPoIB, 62

L

limitpriv プロパティ, 48
linkmode, 60
lofi デバイス
 リムーバブル, 35

N

net リソース
 共有 IP ゾーン, 29
 排他的 IP ゾーン, 30
npiv プロパティ, 52
npiv リソース
 solaris ブランド, 21

P

pagesize-policy, 21

pkey, 60, 62
pool プロパティ, 52

R

RDS (Reliable Datagram Sockets), 32
rootzpool リソース
 solaris ブランド, 22

S

scheduling-class プロパティ, 49
SSM (ADI)
 カーネルゾーン, 26
sys_time, 48

V

virtual-cpu リソース, 49
 カーネルゾーン, 17
VLAN, 62

Z

ZFS
 データセット, 50
zone.cpu-cap リソース制御, 39
zone.cpu-shares リソース制御, 39
zone.max-locked-memory リソース制御, 40
zone.max-lofi リソース制御, 40
zone.max-lwps リソース制御, 40
zone.max-msg-ids リソース制御, 40
zone.max-processes リソース制御, 40
zone.max-sem-ids リソース制御, 40
zone.max-shm-ids リソース制御, 40
zone.max-shm-memory リソース制御, 40
zone.max-swap リソース制御, 40
zonecfg
 admin 承認, 15
 autoboot, 14
 global-time, 14
 template, 11
 一時プール, 16

エンティティ, 46
サブコマンド, 43
操作, 13
大域ゾーン内, 42
モード, 43
有効範囲, 43
有効範囲、大域, 43
有効範囲、リソース固有, 43
zones
 tokens, 11
zpool リソース, 24

