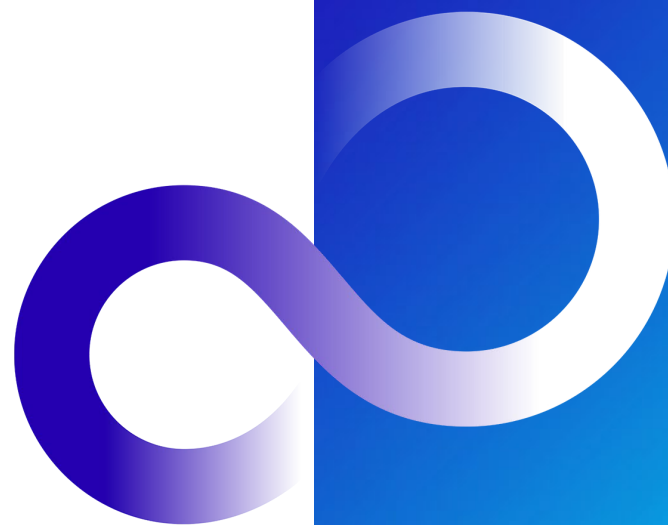


Oracle Solaris 11 ゾーンを 使ってみよう (構築・運用ガイド)

2019年6月 (第4.0版)

富士通株式会社



■ 目的

- 本書は、Oracle Solaris 11 ゾーンの構築・運用方法を紹介します。

■ 対象読者

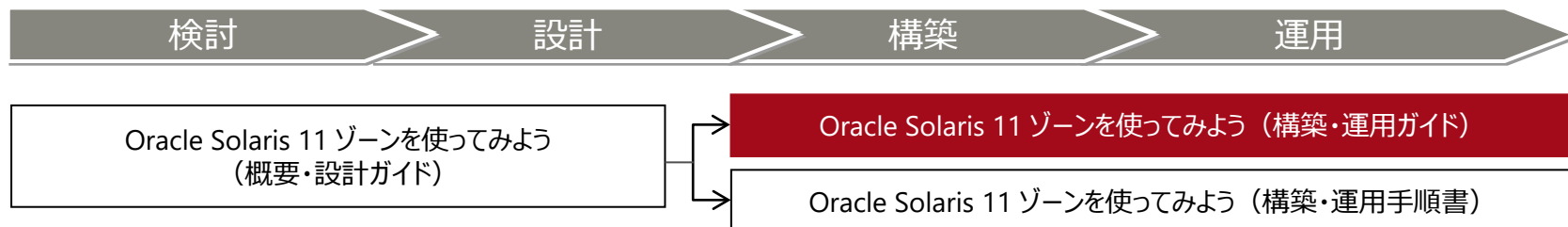
- Oracle Solaris 11の基礎知識を有している方
- 『Oracle Solaris 11を使ってみよう』、『Oracle Solaris 11 ゾーンを使ってみよう（概要・設計ガイド）』を参照された方

■ 留意事項

- 本書の内容は、Oracle Solaris 11.4に基づいています。
Oracle Solaris 11に関する最新情報は、Oracle社のマニュアルをご参照ください。
 - Oracle Solaris Documentation
<https://docs.oracle.com/en/operating-systems/solaris.html>
- 本書では、カーネルゾーンの情報は記載していません。

■ ドキュメントの位置づけ

- Oracle Solaris 11 ゾーンを使ってみよう
<https://www.fujitsu.com/jp/sparc-technical/document/solaris/#solaris-zone>



※ 本書は、『Oracle Solaris 11 ゾーンを使ってみよう（構築・運用手順書）』とあわせてお読みください。

■ 本書での表記

- コマンドのセクション番号は省略しています。

例：

- ls(1) ⇒ lsコマンド
- shutdown(1M) ⇒ shutdownコマンド

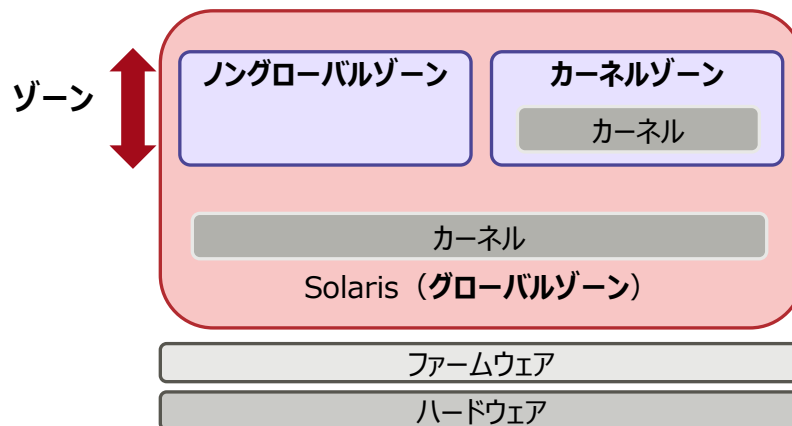
- 以下の用語は略称を用いて表記する場合があります。

略称	正式名称
Solaris	Oracle Solaris
Solarisゾーン	Oracle Solarisゾーン
Oracle VM	Oracle VM Server for SPARC
OBP	Open Boot Prom
IPS	Image Packaging System
GDS	PRIMECLUSTER GD
GLS	PRIMECLUSTER GL

■ 本書で使用するOracle Solarisゾーン関連の用語

- **Solarisゾーン**とは、Solarisの仮想環境を提供するためのサーバ仮想化機能の1つです。
- Solarisゾーンを用いて作成できるSolarisの仮想環境を**ゾーン**と呼びます。
- ゾーンには、**ノングローバルゾーン**と**カーネルゾーン**の2種類があります。

ゾーンの種類	説明
ノングローバルゾーン	グローバルゾーン（*1）とカーネルを共有する仮想環境です。 ※ドキュメントによっては、「non-global zone」と記載しています。
カーネルゾーン	ゾーンごとに独立したカーネルを持つ仮想環境です。 ※Solaris 11.2からサポートされています。

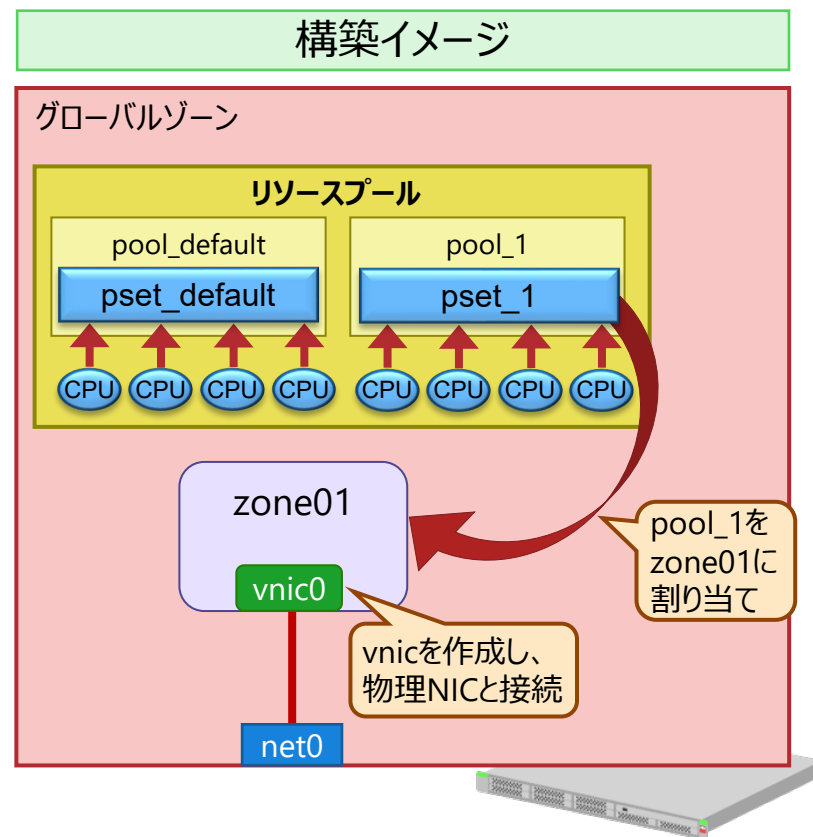
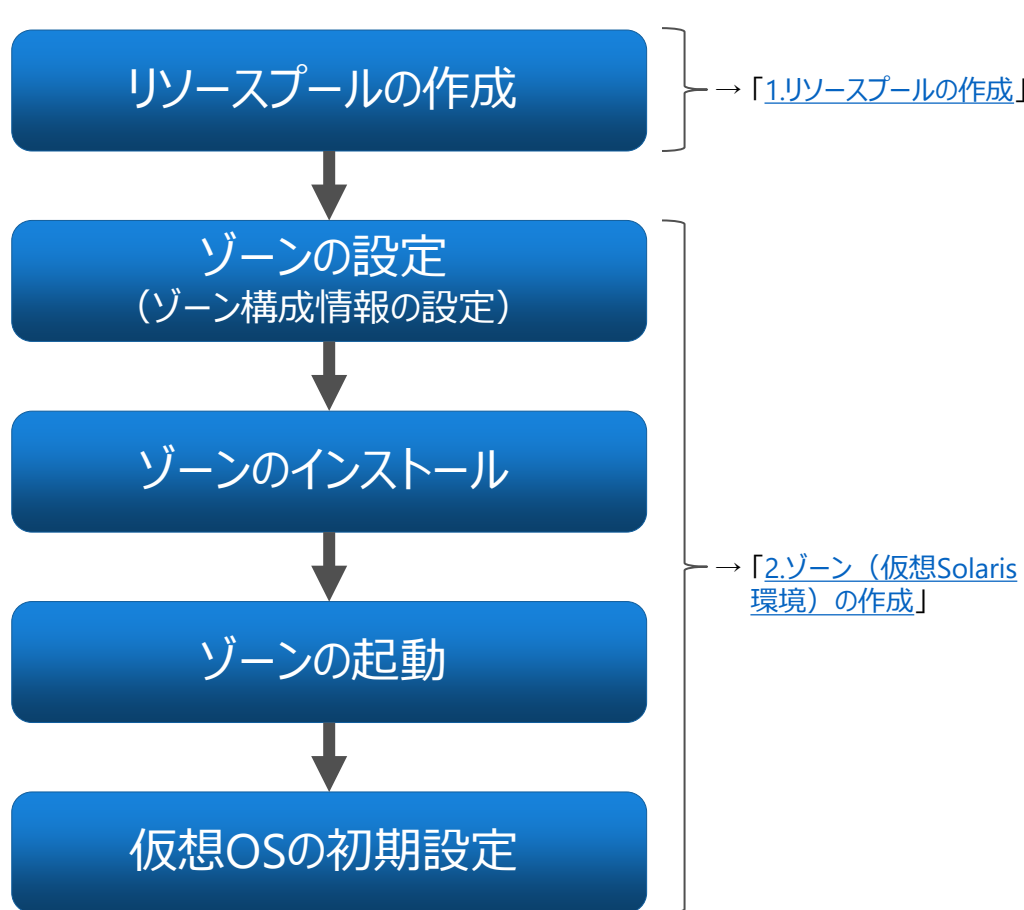


*1 : **グローバルゾーン**とは、物理サーバ上で動作するOS環境です。グローバルゾーンから、ゾーンの設定や制御をします。



• 本書では、「ノングローバルゾーン」を「ゾーン」または「zone」と表記している箇所があります。

1. リソースプールの作成
 2. ゾーン(仮想Solaris環境)の作成
 3. ゾーン(仮想Solaris環境)の基本操作
 4. ゾーン(仮想Solaris環境)の複製
 5. ゾーン(仮想Solaris環境)の移動
 6. ゾーン(仮想Solaris環境)のCPUリソース操作
 7. ゾーン(仮想Solaris環境)の仮想ネットワークの応用
- 付録



- ゾーン構築以外の基本的な操作や運用時の操作については、「[3.ゾーン（仮想Solaris環境）の基本操作](#)」～「[7.ゾーン（仮想Solaris環境）の仮想ネットワークの応用](#)」を参照してください。
- リソースプールを作成しなくても、ゾーンを構築できます。リソースプールを作成しない場合は、すべてのCPUリソースをグローバルゾーンと共有します。
- 本書では、ゾーンのネットワークについてはvnicを使用する方式を中心に記載しています。

1.リソースプールの作成

ゾーン専用のCPUリソースを割り当てる場合は、リソースプールを作成します。
ここでは、リソースプール作成のながれと作成手順を説明します。

ゾーンを作成する準備として、リソースプールを作成します。
リソースプールを作成することで、ゾーンにCPUリソースを割り当てることができます。

■ リソースプール作成のながれ

1.スケジューラの設定

グローバルゾーンのスケジューラを変更します。

2.サービスの起動

リソースプールサービスを起動します。

3.リソースプールの設定

リソースプール構成ファイルを作成・編集し、リソースプールの設定を行います。

4.リソースプールの作成

リソースプール構成ファイルを元に、リソースプールを作成します。

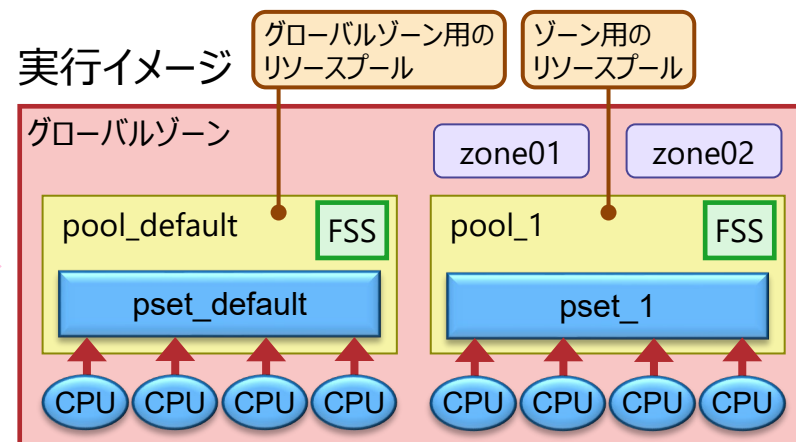


- リソースプール作成時に設定するスケジューラは、リソースプールのCPUリソース配分方式です。
以下の2つの方式があります。
 - FSS (Fair Share Scheduler) :
ゾーンごとにCPU使用率を設定し、CPUリソースを配分します。
特定のプロセスに集中的にCPUリソースを割り当てることができます。
 - TS (Time Share Scheduler) :
CPUリソースをプロセスごとに均等に割り振ります。

■ リソースプールの設定項目と設定例

リソースプール設定項目

	グローバルゾーン用	ゾーン用
リソースプール	pool_default	pool_1
スケジューラ	FSS	FSS
プロセッサセット名	pset_default	pset_1
最大CPUスレッド数	65535	4
最小CPUスレッド数	4	4



■ リソースプール作成に使用するコマンド

コマンド	説明
priocntl	プロセスごとのスケジューラの設定を行います。
dispadmin	デフォルトのスケジューラを変更します。
pooladm	- リソースプール構成ファイル (/etc/pooladm.conf) を作成します。 - poolcfgコマンドで定義した「リソースプール構成」を適用します。
poolcfg	リソースプール構成ファイル (/etc/pooladm.conf) に、「リソースプール設定項目」で設定した内容を「リソースプール構成」として定義します。



リソースプールに割り当てるCPU数の範囲

- 最大CPUスレッド数は、リソースプールに割り当てられるCPUスレッドの最大数を表します (pset.maxプロパティ)。
- 最小CPUスレッド数は、リソースプールに割り当てられるCPUスレッドの最小数を表します (pset.minプロパティ)。

■ スケジューラの設定

- デフォルトのスケジューラをTSからFSSへ変更します。
- OSを再起動すると、変更が反映されます。

✓ デフォルトのスケジューラの変更

【書式】 `dispadmin -d <スケジューラ名>`

```
# dispadmin -d FSS
```

✓ OSの再起動

【書式】 `shutdown -y -g <シャットダウンまでの時間（秒）> -i <コマンド実行後のランレベル>`

```
# shutdown -y -g0 -i6
```

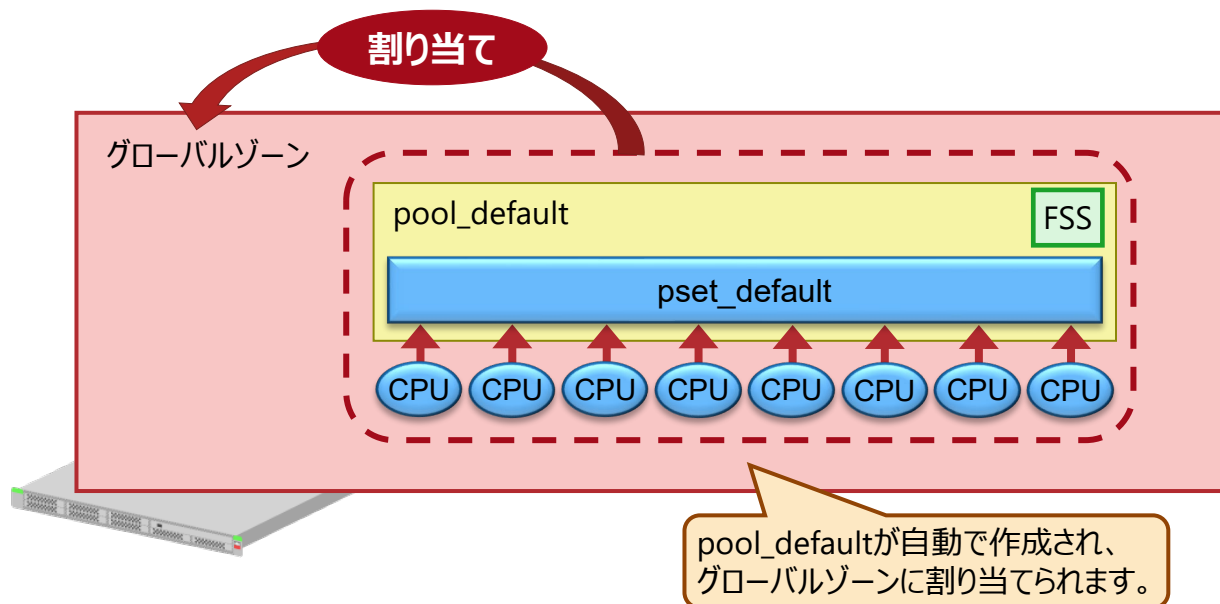
■ サービスの起動

- リソースプールのサービスを起動します。
- サービスを起動すると、デフォルトのリソースプールが作成され、グローバルゾーンにすべてのリソースが割り当てられます。

✓ リソースプールサービスの起動

【書式】 `svcadm enable <サービス名>`

```
# svcadm enable svc:/system/pools:default
```



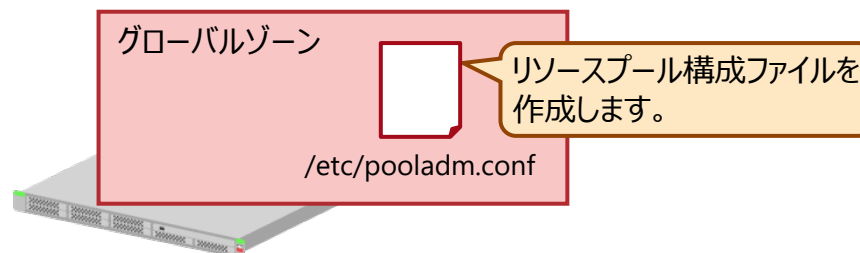
■ リソースプールの設定

- リソースプール構成ファイル（/etc/pooladm.conf）を作成・編集し、リソースプール構成を定義します。

- ✓ リソースプール構成ファイルを作成（pooladmコマンド）

【書式】 pooladm -s

```
# pooladm -s
```



リソースプール構成の定義内容の設定については、次ページ「[リソースプールの作成手順 4/5](#)」を参照してください。

- 例として、右の表に示したリソースプール設定項目をリソースプール構成として定義します。

リソースプール設定項目	グローバルゾーン用	ゾーン用
リソースプール	pool_default	③④ pool_1
スケジューラ	FSS	⑤ FSS
プロセッサセット名	pset_default	②④ pset_1
最大CPUスレッド数	65535	② 4
最小CPUスレッド数	① 4	② 4

- ✓ リソースプール構成の定義（poolcfgコマンド）

【書式】 poolcfg -c <コマンド>

- ① デフォルトのプロセッサセットの最小CPUスレッド数を変更

```
# poolcfg -c 'modify pset pset_default (uint pset.min=4)'
```

- ② 新規プロセッサセット（pset）の作成（最大CPUスレッド数と最小CPUスレッド数は4を設定）

```
# poolcfg -c 'create pset pset_1 (uint pset.min=4; uint pset.max=4)'
```

- ③ 新規リソースプール（pool）の作成

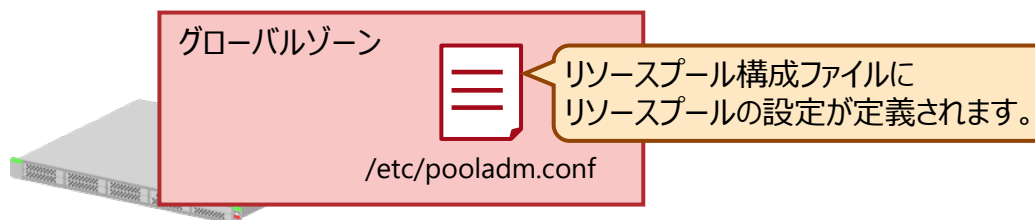
```
# poolcfg -c 'create pool pool_1'
```

- ④ プロセッサセットとリソースプールの関連付け

```
# poolcfg -c 'associate pool pool_1 (pset pset_1)'
```

- ⑤ リソースプールのスケジューラ設定

```
# poolcfg -c 'modify pool pool_1 (string pool.scheduler="FSS")'
```



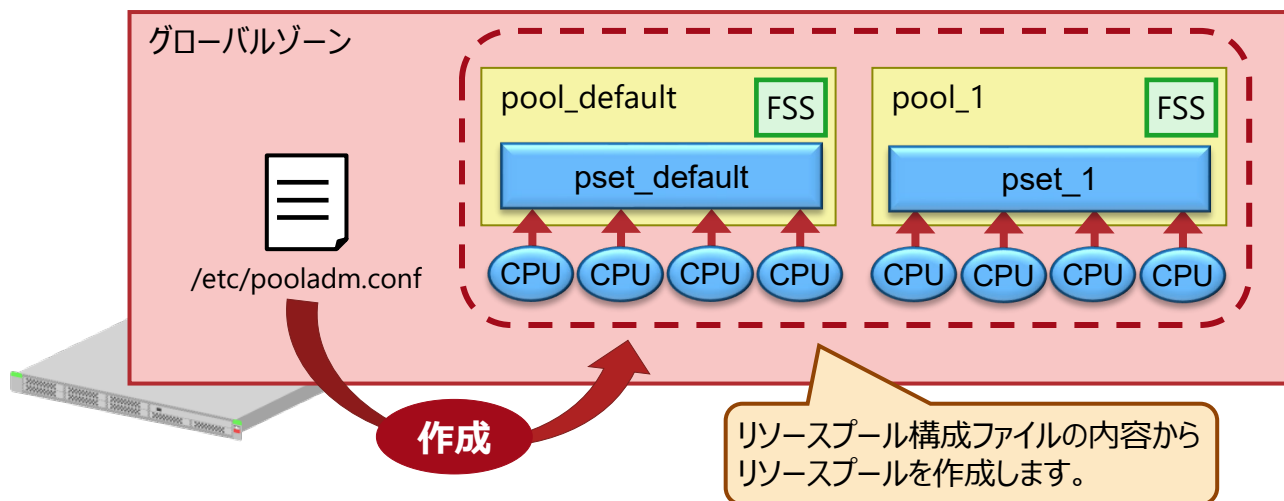
■ リソースプールの作成

- リソースプール構成ファイルの内容を元にリソースプールを作成します。

✓ リソースプールを作成（pooladmコマンド）

【書式】 pooladm -c

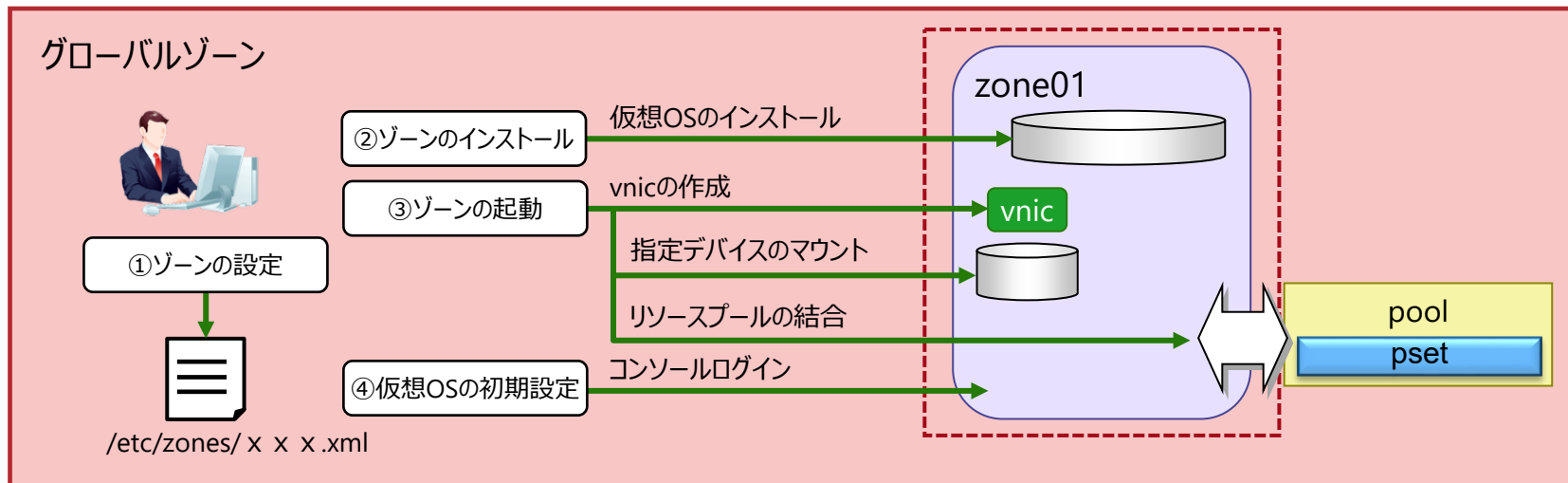
```
# pooladm -c
```



2.ゾーン（仮想Solaris環境）の作成

ゾーン作成のながれと作成手順を説明します。

■ ゾーン作成のながれ



① ゾーンの設定

ゾーンの構成情報（ゾーン設定ファイル）を作成します。

② ゾーンのインストール

ゾーンをインストールします。

③ ゾーンの起動

ゾーンを起動します。このとき、仮想ネットワークインターフェース（vnic）は自動で作成されます。

④ 仮想OSの初期設定

対話形式でホスト名、パスワード、言語などを設定します。

■ ゾーン作成に使用するコマンド

- zonecfgコマンド
ゾーンの構成情報（ゾーン構成ファイル）を作成します。
- zoneadmコマンド
ゾーンをインストールし、運用（起動／停止／再起動）します。
- zloginコマンド
ゾーンへログインします。

■ ゾーンの構成

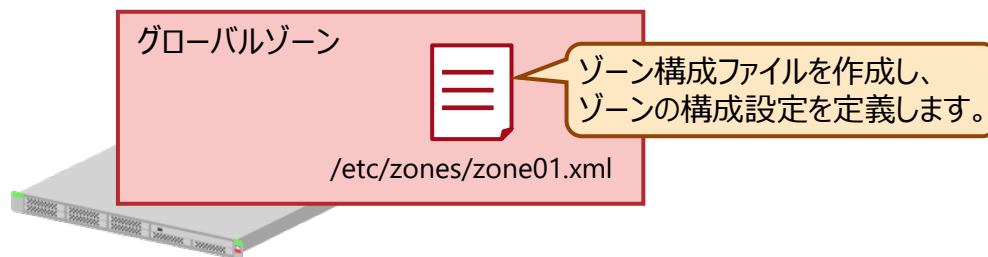
- ✓ ゾーン構成の設定 (zonecfgコマンド)

【書式】 zonecfg -z <ゾーン名>

```
# zonecfg -z zone01
zonecfg:zone01> create
zonecfg:zone01> set zonepath=/zones/zone01
zonecfg:zone01> set autoboot=false
zonecfg:zone01> select anet lower-link=auto
zonecfg:zone01:anet> set lower-link=net0
:
zonecfg:zone01> exit
```

ゾーンの構成を定義

→ ゾーンの設定項目について詳しくは、「[《参考》ゾーン構成情報の設定項目](#)」を参照してください。



- ゾーン構成の設定、または変更時の実行例を記載します。
- 本書では、リソース指定の終了 (end) やセッションの終了 (exit) などを省略している場合があります。

```
# zonecfg -z zone01
zonecfg:zone01> select anet lower-link=auto
zonecfg:zone01:anet> set lower-link=net0
zonecfg:zone01:anet> end
zonecfg:zone01> exit
```

←selectで該当のリソースを選択 (必要に応じて)
←setでプロパティの値を入力
←endでリソースを閉じる (select実行時のみ必要)

■ ゾーンのインストール

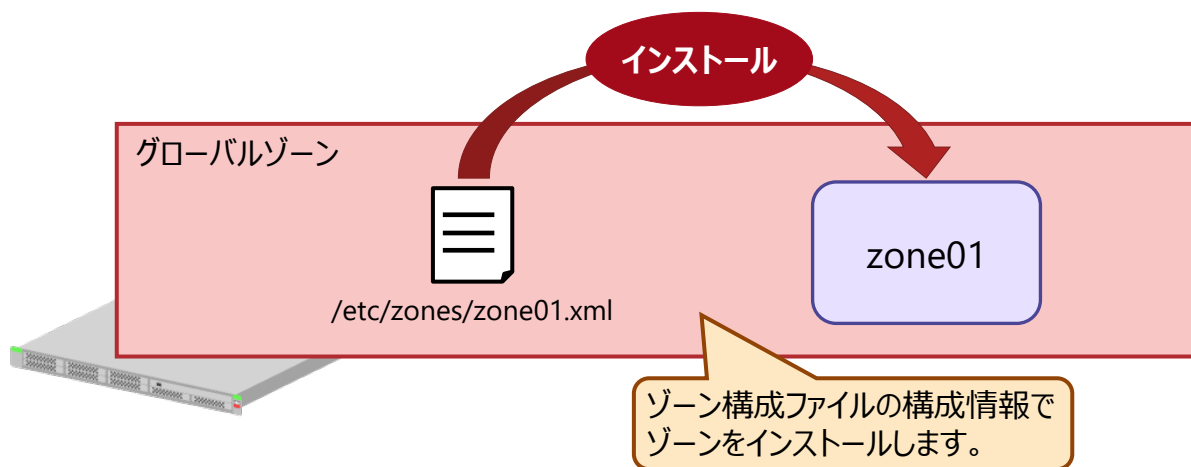
- 作成したゾーン構成ファイルの内容で、ゾーンがインストールされます。

✓ ゾーンのインストール（zoneadmコマンド）

【書式】 zoneadm -z <ゾーン名> install

```
# zoneadm -z zone01 install
```

※グローバルゾーン内のローカルリポジトリ、またはリポジトリサーバからインストールします。



• ゾーンをインストールする際は、リポジトリサーバが構成済みである必要があります。リポジトリサーバの詳細については、以下を参照してください。

- Oracle Solaris 11を使ってみよう

<https://www.fujitsu.com/jp/sparc-technical/document/solaris/#os>

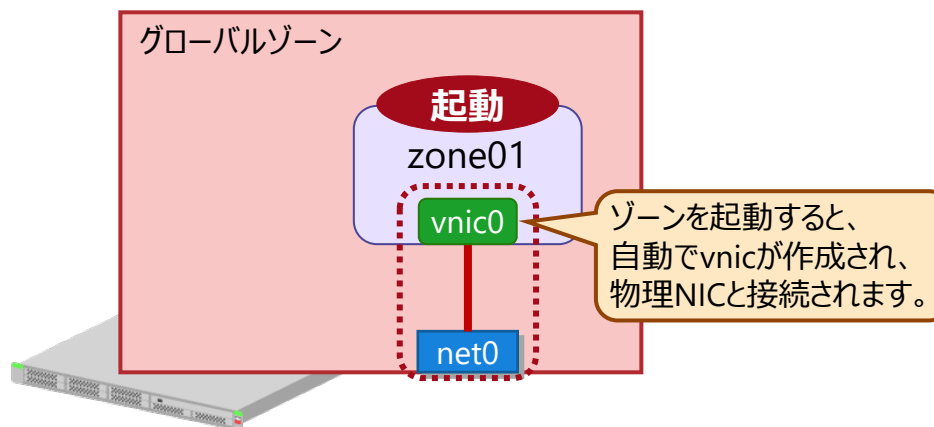
■ ゾーンの起動

- ゾーンを起動します。このとき、仮想ネットワークインターフェース（vnic）は自動で作成されます。

✓ ゾーンの起動（zoneadmコマンド）

【書式】 zoneadm -z <ゾーン名> boot

```
# zoneadm -z zone01 boot
```



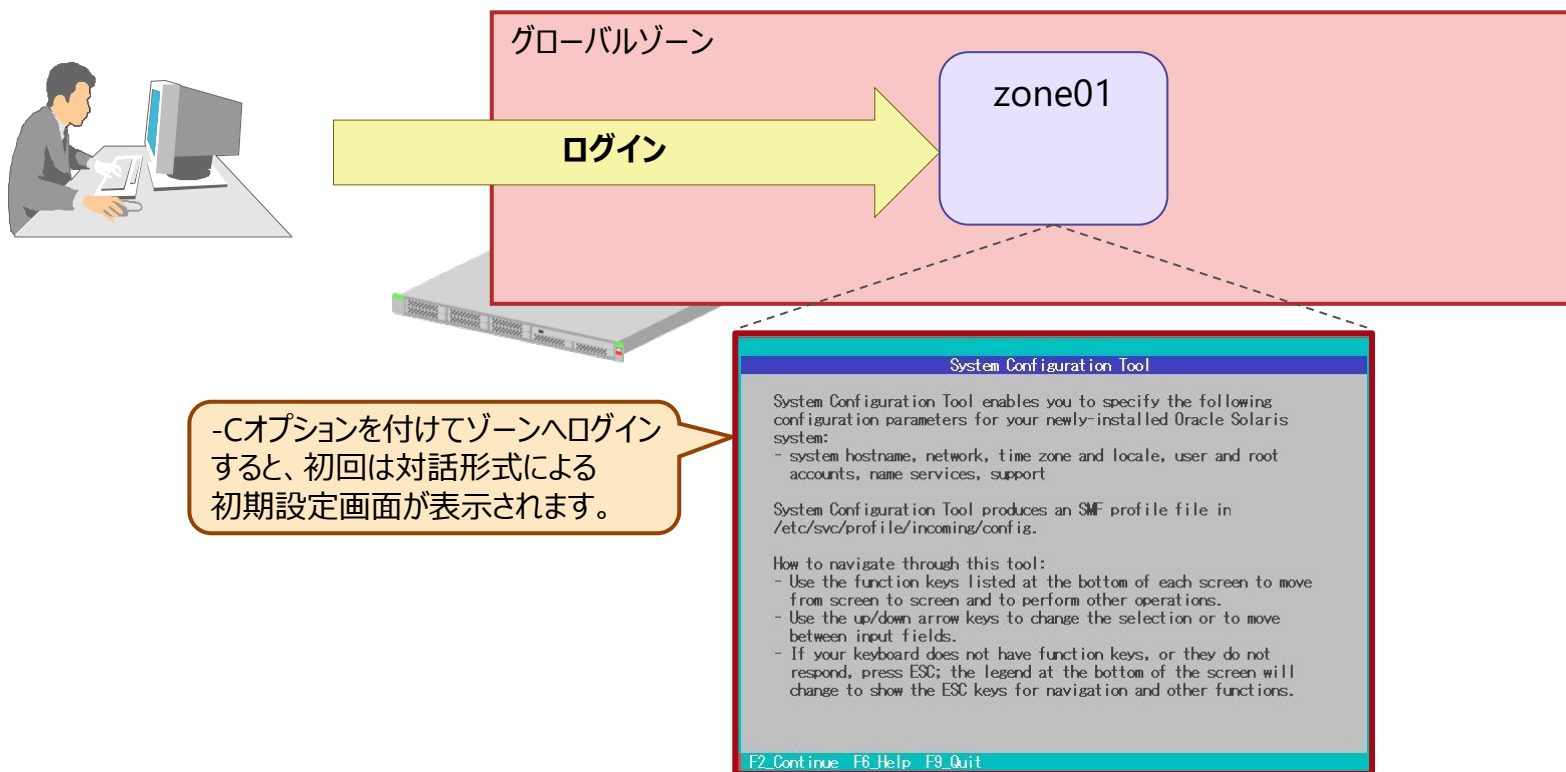
■ 仮想OSの初期設定

- 対話形式でホスト名、パスワード、言語などを設定します。

✓ ゾーンへログイン、コンソール接続 (zloginコマンド)

【書式】 zlogin -C <ゾーン名>

```
# zlogin -C zone01
```



• -Cオプションを付けずに、初期設定画面は表示されません。

《参考》ゾーン構成情報の設定項目 1/2

設定項目 ◎：必須 ※：排他			設定値（例）	説明
ゾーン名	◎	zonename		ゾーンの名前
ゾーンパス	◎	zonepath	/zones/zone01	ゾーンがインストールされるパス
起動オプション		bootargs	-m verbose	永続的なゾーンの起動引数
自動起動		autoboot	true	グローバルゾーンと同時起動
ブランドタイプ		brand	solaris	ゾーンの属性
特権		limitpriv	default,sys_time	ゾーンに付与する特権
スケジューラ	◎	scheduling-class	FSS	スケジューリングクラス
リソースプール	※	pool	pool_1	ゾーン起動時にバインドするリソースプールの名前
CPU占有	※	dedicated-cpu	ncpus	3 ゾーン専用CPUの割り当て →「 《参考》CPUの占有設定 」参照
メモリ制限		capped-memory	physical	2G
			swap	4G
			lock	1G
CPU制限	※	capped-cpu	ncpus	1 CPUリソースの上限値 →「 CPUキャップの設定 」参照
IPタイプ	◎	ip-type	exclusive	ネットワーク接続構成 →「 《参考》ゾーンのネットワーク構成設定 」参照
ネットワーク	◎	anet	linkname	vnici0
			lower-link	net0
			configure-allowed-address	true
			link-protection	mac-nospoof
			mac-address	auto

《参考》ゾーン構成情報の設定項目 2/2

設定項目 ◎：必須 ※：排他			設定値（例）	説明
共有ファイルシステム	fs	dir	/mnt	ゾーン側のマウントポイント
		special	/work	グローバルゾーン側の共有ディレクトリ
		type	lofs	ファイルシステムのタイプ
		(option)	ro	マウントオプション
共有デバイス	device	match	/dev/rmt/*	ゾーンから直接のアクセスを許可するデバイス
リソース制御	※	cpu-shares	1	スケジューラ「FSS」のCPUシェア数
		max-lwps	100	同時に使用できる LWP の最大数
		max-msg-ids	512	メッセージキューIDの最大数
		max-processes	10	同時に使用できるプロセステーブルスロットの最大数
		max-sem-ids	256	セマフォIDの最大数
		max-shm-ids	256	共有メモリIDの最大数
		max-shm-memory	1G	共有メモリの最大数
コメント	attr	name	comment	コメント名
		type	string	コメントのデータ型
		value	"This is DB Server"	コメント内容



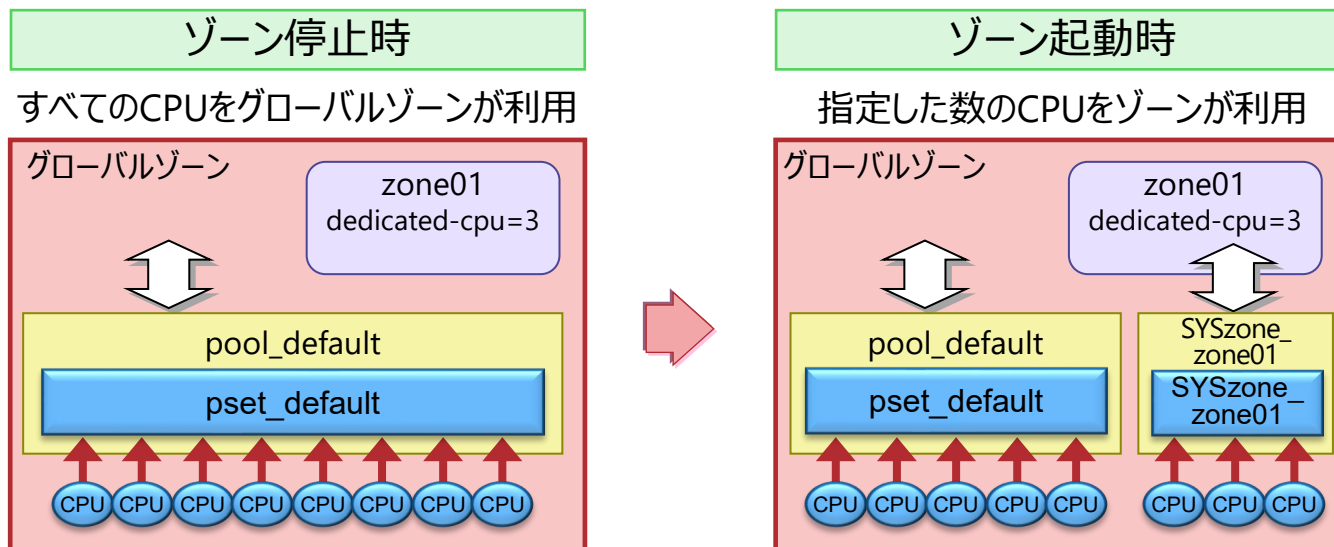
・ゾーン構成情報の設定手順については、「[ゾーンの構成情報の変更](#)」を参照してください。

CPUの占有設定は、ゾーンに対する以下の割り当てを解除してから実施してください。

- リソースプール（poolプロパティ）
- CPUシェア数の設定（cpu-sharesプロパティ）

■ dedicated-cpuリソース（ゾーン専用CPU設定）

- ゾーンに割り当てるCPU数を指定できます。
- ゾーンを起動すると、指定したCPU数で専用のリソースプールを構成します。



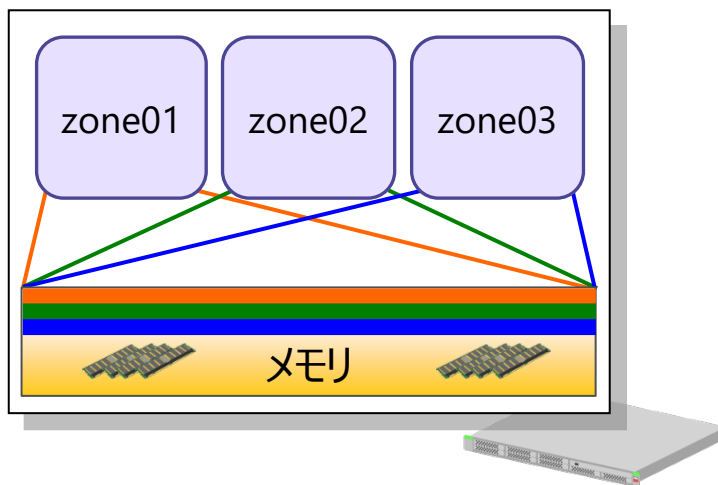
- CPU数の範囲指定が可能です（例：dedicated-cpu=1-4）。
- dedicated-cpuリソースを設定したゾーンの起動時に、指定されたCPUリソースが確保できない場合、ゾーンを起動できません。dedicated-cpuで設定した最小CPU数と、ほかのプロセッサセット（pset）で設定した最小CPU数の合計が、物理CPU数を超えないように設定してください。
- 複数ゾーン間でリソースプールを共有する場合は、poolcfgコマンドでリソースプールを作成して各ゾーンに割り当てます。

■ capped-memoryリソース（メモリ制限設定）

- ・ゾーンに対するメモリ使用制限が可能です。

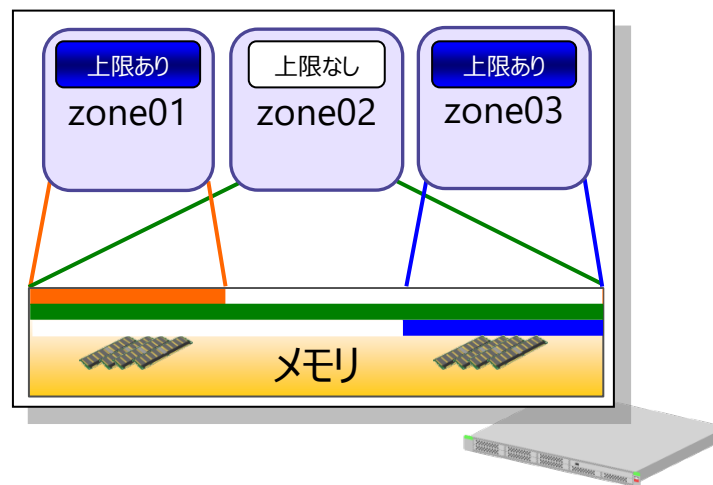
メモリ制限なし

- ・メモリ領域は共有のため、各ゾーンは必要とするメモリを使用します。
- ・メモリ不足の場合、ミドルウェアやアプリケーションの動作に影響を与えます。



メモリ制限あり

- ・ゾーン単位で、以下のメモリ資源の使用上限を設定できます。
physical : 物理メモリ
swap : スワップ領域
locked : ロックされた物理メモリ



- ・ゾーン単位でメモリ資源の使用上限を設定することで、特定のゾーンがメモリを占有することによる、ほかのゾーンの性能劣化を回避できます。
- ・本番環境と検証環境が混在するような構成では、メモリリソースの制限を有効活用できます。

■ ip-typeプロパティ (IPタイプ)

ip-typeで、ゾーンのネットワーク構成を設定します。

- ip-typeは、「shared」または「exclusive」のどちらかに設定します。

※ Solaris 11では、「exclusive」(占有) がデフォルト設定です。

■ shared (① ネットワーク単純共有型)

- 単一の物理LANポート(物理NIC)を共有する構成です。

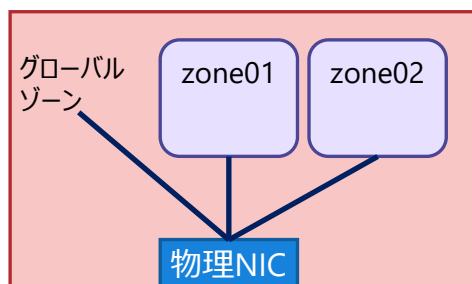
■ exclusive (② ネットワーク占有型)

- 物理LANポートを特定のゾーンに占有させる構成です。

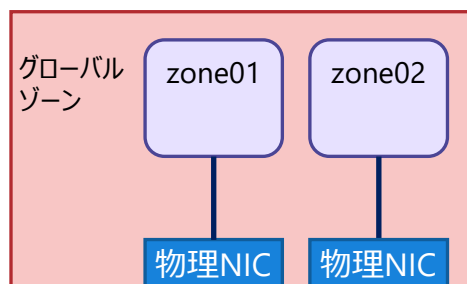
■ exclusive + vnic (③ ネットワーク仮想共有型)

- ネットワーク占有型と同様にexclusiveの方式ですが、「vnic」と呼ばれる仮想ネットワークデバイスを使用することで、単一の物理LANポートを共有することが可能となった構成です。

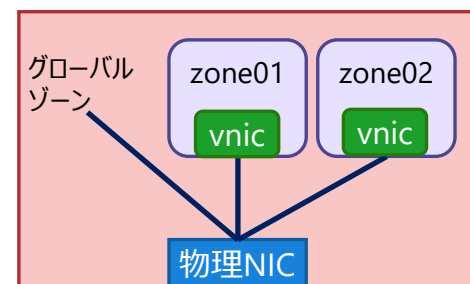
①ネットワーク単純共有型



②ネットワーク占有型



③ネットワーク仮想共有型



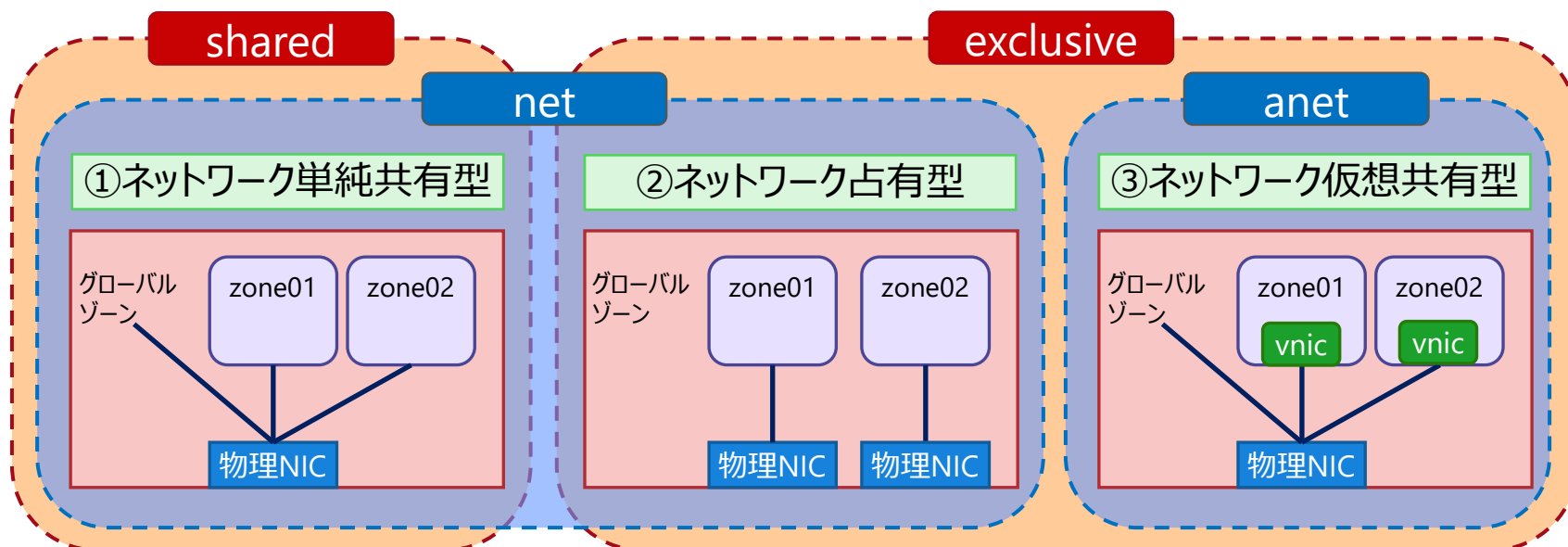
・本書は、最も高機能でオペレーションが簡単な「③ネットワーク仮想共有型」を中心に記載しています。

■ プロパティと実行コマンドの違い

- Solaris ゾーンのネットワーク構成の割り当て方法①～③は、zonecfgコマンドのプロパティ、および以下の実行コマンドで指定します。

ネットワークの割り当て方法	プロパティ : ip-type	実行コマンド
① ネットワーク単純共有型	shared	add net
② ネットワーク占有型	exclusive	add net
③ ネットワーク仮想共有型	exclusive	add anet (※1)

※1 : add netコマンドでも設定できますが、オペレーションが簡単で仮想化構成もシンプルなadd anetを推奨します。



■ 仮想ネットワークデバイス（vnic）の概要

■ リソース

- anet
 - デフォルト設定で、anetを使用した占有設定（ip-type=exclusive）で構成されます。
 - ゾーンの起動時に自動的にvnicが作成され、ゾーン停止時に削除されます。
- net
 - vnicを事前に作成し、ゾーン構成時に割り当てる仮想NICを指定します。
 - ゾーンの起動／停止を問わず、vnicの設定情報は削除されません。

■ プロパティ

- anet

下記のプロパティがデフォルトで構成されます。

プロパティ	説明	デフォルト設定
linkname	ゾーン上で認識される仮想NICのデバイス名	net0
lower-link	仮想NICと紐づく物理NICの設定	auto
configure-allowed-address	初回起動時のallowed-address（※1）を永続的に構成するか否か	true
link-protection	リンク保護のタイプの指定	mac-nospoof
mac-address	仮想NICのMACアドレスの指定	auto

※1：ゾーンに対して許可するIPアドレス

- net

プロパティ	説明
physical	ゾーンに割り当てるnicの指定

■ ゾーンの削除手順

■ ゾーン環境の削除

① ゾーンの停止

【書式】 zoneadm -z <ゾーン名> shutdown

```
# zoneadm -z zone01 shutdown
```

② ゾーンのアインストール

【書式】 zoneadm -z <ゾーン名> uninstall

```
# zoneadm -z zone01 uninstall
```

③ ゾーンの削除

【書式】 zonecfg -z <ゾーン名> delete

```
# zonecfg -z zone01 delete
```

■ リソースプール機能の停止

① リソースプールの初期化

【書式】 pooladm -x

```
# pooladm -x
```

② リソースプールサービスの停止

【書式】 svcadm disable <サービス名>

```
# svcadm disable pools:default
# svcadm disable pools/dynamic:default
```

3.ゾーン（仮想Solaris環境）の基本操作

ゾーンの起動／停止、ログインなどの基本操作を説明します。

■ ゾーンの基本操作と使用するコマンド

ゾーンの基本操作	コマンドまたはオプション
<u>ゾーンのステータスの確認</u>	zoneadmコマンド
<u>ゾーンの起動／停止</u>	zoneadmコマンド
<u>ゾーンへのログイン</u>	zloginコマンド
<u>ゾーンへのコマンド実行</u>	zloginコマンド
<u>ゾーンの情報の確認</u>	-Zオプション
<u>ゾーンのネットワークの管理</u>	ipadmコマンド
<u>ゾーンにおける制限と特権の設定</u>	zonecfgコマンド
<u>ゾーンの構成情報の変更</u>	zonecfgコマンド
<u>ゾーンのディレクトリの変更（サーバ内）</u>	zoneadmコマンド
<u>ゾーンの統計情報の確認</u>	zonestatコマンド



• ゾーン環境でも、物理環境（グローバルゾーン）と同様にBE（Boot Environment）を利用できます。BEについて詳しくは、以下を参照してください。

- Oracle Solaris 11を使ってみよう

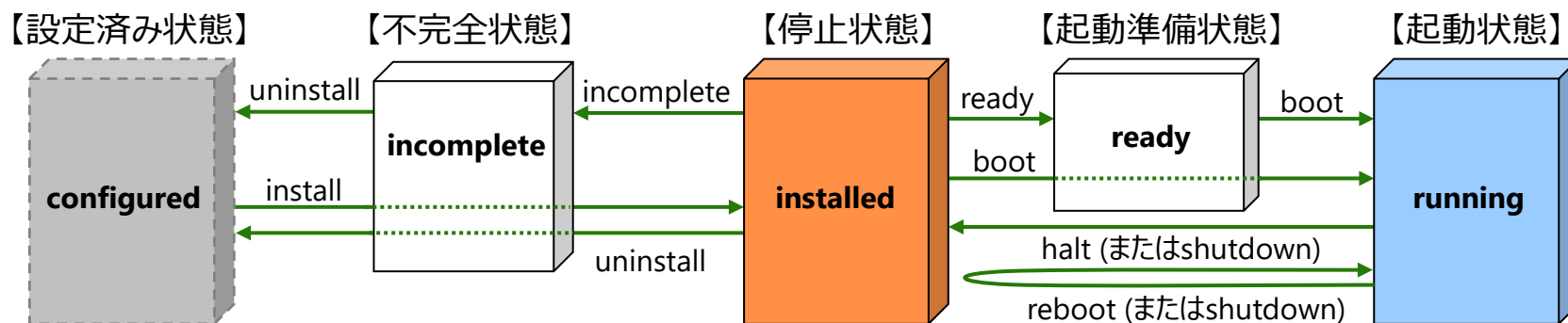
<https://www.fujitsu.com/jp/sparc-technical/document/solaris/#os>

- ゾーンの状態（設定済み、起動準備中、起動中、停止中など）を確認します。

- ✓ ゾーンの状態確認

【書式】 zoneadm list [オプション]

```
# zoneadm list -vc
ID NAME          STATUS          PATH          BRAND          IP
0  global         running         /              solaris         shared
2  zone01         ready           /zones/zone01 solaris         excl
-  zone02         installed       /zones/zone02 solaris         excl
-  zone03         incomplete     /zones/zone03 solaris         excl
-  zone04         configured     /zones/zone04 solaris         excl
```



• ゾーンのインストールの中断または失敗により、「不完全状態 (incomplete)」になる場合があります。その際は、ゾーンをアンインストールしてください。

- 通常運用時にゾーンを起動、停止、または再起動したり、緊急時に再起動または停止したりします。

- ✓ 通常運用時のゾーンの起動、再起動、停止

- 【書式】 zoneadm -z <ゾーン名> [boot | shutdown]

- 【書式】 zlogin <ゾーン名> <コマンド>

- ゾーンの起動

- ```
zoneadm -z zone01 boot
```

- ゾーンの停止

- ```
# zoneadm -z zone01 shutdown
```

- ゾーンの再起動

- ```
zlogin zone01 'shutdown -y -g0 -i6'
```

- ✓ 緊急時のゾーンの再起動、停止（※ゾーンを強制的に停止または再起動します）

- 【書式】 zoneadm -z <ゾーン名> [reboot | halt]

- ゾーンの再起動

- ```
# zoneadm -z zone01 reboot
```

- ゾーンの停止

- ```
zoneadm -z zone01 halt
```

- ✓ シングルユーザーモードで起動

- 【書式】 zoneadm -z <ゾーン名> boot -s

- ```
# zoneadm -z zone01 boot -s
```

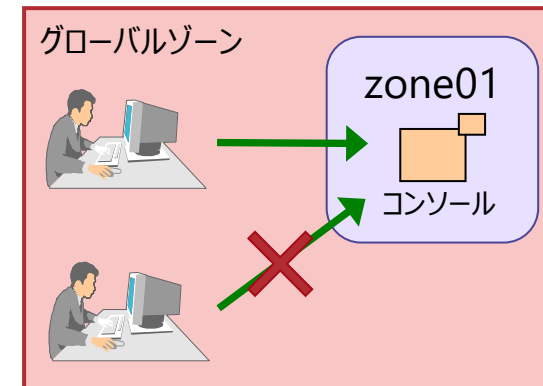
■ グローバルゾーンからゾーンにログインします。

✓ コンソールに接続

ゾーン停止中でも接続できます。同時に1ユーザーのみ接続可能です。
OSインストール直後に初期設定を実施する場合や、ゾーンの再起動を伴う作業を実施する場合は、コンソールに接続します。

【書式】 `zlogin -C <ゾーン名>`

```
# zlogin -C zone01
[Connected to zone 'zone01' console]
```

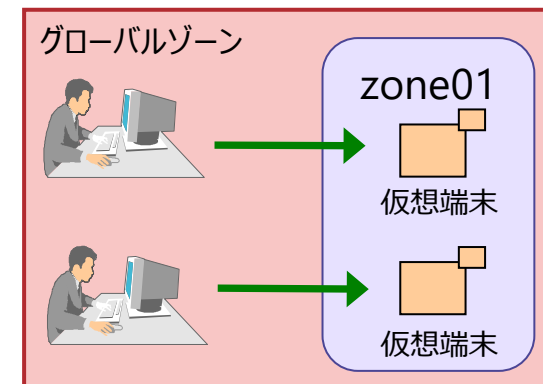


✓ 仮想端末に接続

ゾーン起動中のみ接続できます。同時に複数ユーザーが接続可能です。
ログイン時のパスワード認証はありません。

【書式】 `zlogin <ゾーン名>`

```
# zlogin zone01
[Connected to zone 'zone01' pts/1]
Last login: Mon Apr  1 16:52:00 2019 on pts/1
Oracle Corporation      SunOS 5.11      11.4      January 2019
```



《参考》一般ユーザー権限でのログイン

事前に、操作権限を与えるユーザー名および権限を、ゾーンのadminプロパティに定義する必要があります。

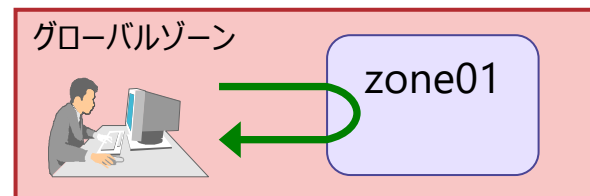
【書式】 `pfexec <コマンド>`

```
$ pfexec zlogin zone01
```

- ゾーンにログインすることなく、グローバルゾーンからゾーン上でコマンドを実行することができます。

- ✓ ゾーン上でのコマンド実行
コマンド実行結果だけが表示されます。

【書式】 `zlogin -l <ユーザー名> <ゾーン名> <コマンド>`



```
# zlogin -l user01 zone01 ps -ef
Oracle Corporation      SunOS 5.11      11.4      January 2019
  UID    PID  PPID   C    STIME TTY          TIME CMD
:
  root   4721  4044   0  13:57:17 ?          0:00 /usr/lib/inet/inetd start
  root   4785  4044   0  13:57:21 ?          0:00 /usr/sbin/syslogd
  user01 4900   4898   0  13:59:01 ?          0:00 ps -ef
:
```

- 各種コマンド実行時、ゾーンに対応したオプション（-z、-Z）を指定することで、グローバルゾーンから各ゾーンの情報を取得できます。

- ✓ プロセス情報を表示

```
# ps -efZ | grep init
global      root      1      0      0 10:07:21 ?           0:00 /usr/sbin/init
zone01      root 10103 10040   0 15:51:04 ?           0:00 /usr/sbin/init
zone02      root 10895 10559   0 15:51:21 ?           0:00 /usr/sbin/init
```

- ✓ ファイルシステム情報を表示

```
# df -Z
:
/zones              (zones              ) : 93568042 blocks 93568042 files
/zones/zone01       (zones/zone01       ) : 93568042 blocks 93568042 files
/zones/zone01/root   (zones/zone01/rpool/ROOT/solaris   ) : 93568042 blocks 93568042 files
/zones/zone01/root/var (zones/zone01/rpool/ROOT/solaris/var) : 93568042 blocks 93568042 files
/zones/zone01/root/dev (/dev              ) :          0 blocks          0 files
:
```



• 上記のほか、ipcsコマンド、pgrepコマンド、pkillコマンド、ptreeコマンド、prstatコマンド、coreadmコマンドなどで-zオプションを指定して実行できます。

■ ネットワークインターフェースの設定

IPアドレスなどのネットワークインターフェースの設定は、ゾーン上で行います。

① ゾーンに割り当てられているネットワークデバイス（主にvnic）の確認

【書式】 `dladm show-link`

```
zone01# dladm show-link
LINK          CLASS      MTU    STATE    OVER
vnic0         vnic       1500   up       ?
```

② IPアドレスの設定

【書式】 `ipadm create-ip <ネットワークデバイス名>`

`ipadm create-addr [-T static] -a local=<IPアドレス> <ネットワークデバイス名>/<識別子>`

```
zone01# ipadm create-ip vnic0
zone01# ipadm create-addr -T static -a local=192.168.10.XX/24 vnic0/v4
```

割り当てられたnicに
対して設定できます。

③ 設定したIPアドレスの確認

【書式】 `ipadm show-addr`

```
zone01# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
vnic0/v4     static    ok         192.168.10.XX/24
lo0/v6       static    ok         ::1/128
```



- ip-typeプロパティが「shared（共有）」に設定されているゾーンについては、グローバルゾーン側でnicを管理します。
- 以降、ゾーン上での操作の場合は、プロンプトを「zone01#」のように「ゾーン名#」で記載します。

■ ゾーン上では動作しないコマンド（一例）

動作しないコマンド（例）	備考
dtrace	dtrace_proc特権、dtrace_user特権を追加することで動作可能。
eeeprom	参照は可能。
format	ディスク/LUN全体をゾーンに委任することで動作可能。 ラベル付けには、allow-partitionプロパティの設定が必要。
prtconf	一部のオプションは実行可能。
prtdiag	—
psradm	—
trapstat	—

■ 特権の設定方法（一例）

- ✓ ゾーンにdtrace関連の特権（dtrace_proc、dtrace_user）を付与

```
zonecfg:zone01> set limitpriv=default,dtrace_proc,dtrace_user
```

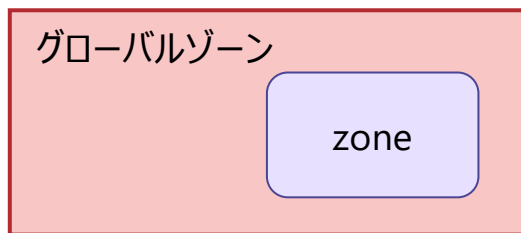


• 基本的に、ゾーンではハードウェアやカーネルを操作するアプリケーションやコマンドは実行できません。
ただし、ゾーンに特権を定義することで、一部の実行権を付与することができます。

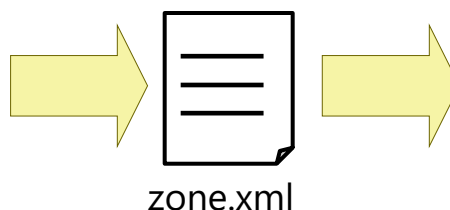
■ ゾーンの作成時に設定した構成情報を変更します。

- ゾーンの構成情報の変更は、ゾーンを停止して行います。変更後、ゾーンを起動します。

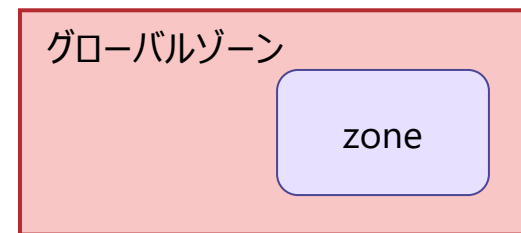
①ゾーンの停止



②zonecfgコマンドで修正

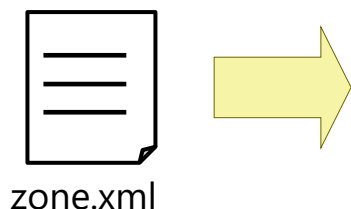


③ゾーンの起動



- 一部の設定項目については、zoneadm apply コマンドで動的に反映することが可能です。
※ゾーンの停止と起動は不要です。

①zonecfgコマンドで修正



②zoneadm applyコマンドで動的に反映



- 動的に設定可能な項目の詳細については、以下のOracle社のマニュアルを参照してください。
https://docs.oracle.com/cd/E37838_01/html/E61040/gqqwk.html#scrolltoc
-Resource Types and Global Properties That Support Live Zone Reconfiguration

■ 仮想NICと紐づく物理NICの設定の変更例（lower-linkプロパティ）

→ ゾーンの定義項目について詳しくは、「[《参考》ゾーン構成情報の設定項目](#)」を参照してください。

① パラメーターの現在の設定値を確認

```
# zonecfg -z zone01
zonecfg:zone01> info -a anet
pool: pool_1
  linkname: vnic0
  lower-link: net0
  allowed-address:
  configure-allowed-address: true
  :
```

lower-linkプロパティのリソースタイプ（anet）を指定します。

現在の設定値が表示されます。

② 設定を変更

【書式】select <リソースタイプ> <プロパティ>=<現在の設定値>

【書式】set <プロパティ>=<変更する値>

```
zonecfg:zone01> select anet lower-link=net0
zonecfg:zone01:anet> set lower-link=net1
zonecfg:zone01:anet> end
zonecfg:zone01> commit
zonecfg:zone01> exit
```

③ 設定を反映

事前にゾーンを停止している場合は、ゾーンを起動すると、変更した構成情報が反映されます。
起動中のゾーンに変更した構成情報を動的に反映する場合は、以下のコマンドを実行します。

【書式】zoneadm -z <ゾーン名> apply

```
# zoneadm -z zone01 apply
Checking: Modifying anet linkname=vnic0
Applying the changes
```


■ 仮想NICと紐づく物理NICの設定のクリア例（lower-linkプロパティ）

→ ゾーンの定義項目について詳しくは、「[《参考》ゾーン構成情報の設定項目](#)」を参照してください。

① パラメーターの現在の設定値を確認

```
# zonecfg -z zone01
zonecfg:zone01> info -a anet
pool: pool_1
  linkname: vnic0
  lower-link: net1
  allowed-address:
  configure-allowed-address: true
  :
```

lower-linkプロパティのリソースタイプ（anet）を指定します。

現在の設定値が表示されます。

② 設定をクリア（クリアするとデフォルト設定に戻ります）

【書式】 select <リソースタイプ> <プロパティ>=<現在の設定値>

【書式】 clear <プロパティ>

```
zonecfg:zone01> select anet lower-link=net0
zonecfg:zone01:anet> clear lower-link
zonecfg:zone01:anet> end
zonecfg:zone01> commit
zonecfg:zone01> exit
```

③ 設定を反映

事前にゾーンを停止している場合は、ゾーンを起動すると、変更した構成情報が反映されます。
起動中のゾーンに変更した構成情報を動的に反映する場合は、以下のコマンドを実行します。

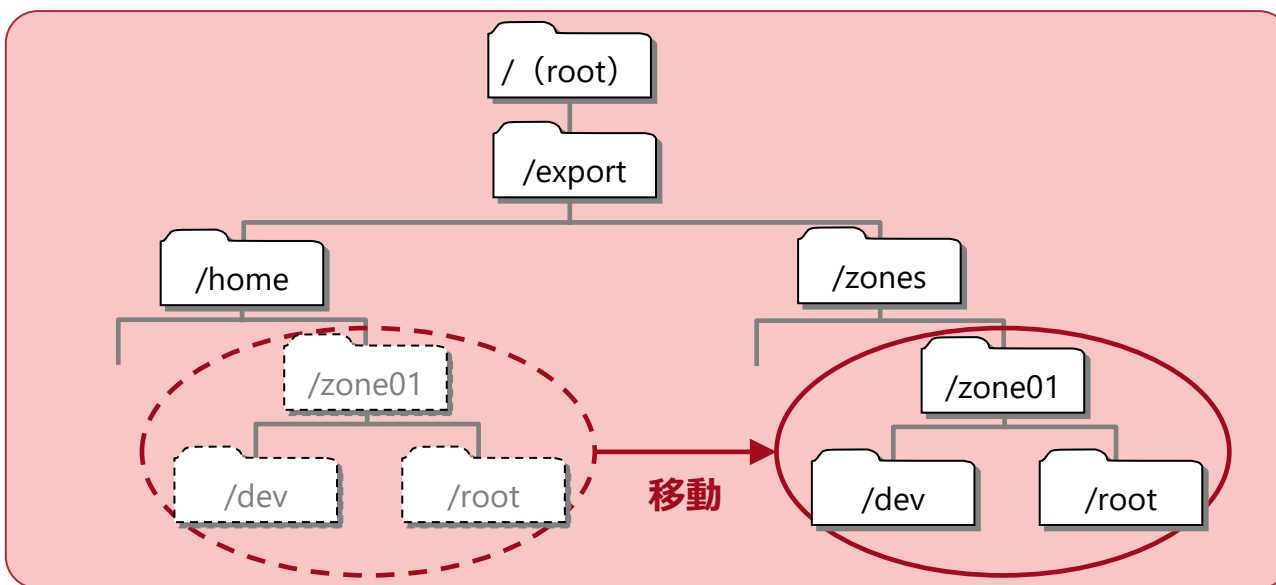
【書式】 zoneadm -z <ゾーン名> apply

```
# zoneadm -z zone01 apply
Checking: Modifying anet linkname=vnic0
Applying the changes
```

ゾーンのディレクトリの変更（サーバ内）

- ゾーンのディレクトリ（zonepath）を、同一サーバ内の別の領域へ変更できます。

例：/export/**home**/zone01 から /export/**zones**/zone01 へ移動



✓ ゾーンのディレクトリの変更

【書式】 zoneadm -z <ゾーン名> move <移動先ディレクトリ>

```
# zoneadm -z zone01 move /export/zones/zone01
```



- ゾーンのディレクトリを変更すると、ゾーン作成時に定義したzonepathプロパティが変更されます。
- 実行前に、移動先のディスク領域が十分であることを確認してください。

■ ゾーンの統計情報を取得できます。

✓ ゾーンの統計情報を取得

【書式】 zonestat <取得間隔 (秒)>

```
global01# zonestat 5
Collecting data for first interval...
Interval: 1, Duration: 0:00:05
SUMMARY
```

	Cpus/Online: 48/48				PhysMem: 125G		VirtMem: 129G			
	-----CPU-----				--PhysMem--		--VirtMem--		--PhysNet--	
ZONE	USED	%PART	STLN	%STLN	USED	%USED	USED	%USED	PBYTE	%PUSE
[total]	0.02	0.04%	0.00	0.00%	9355M	7.27%	10.7G	8.31%	0	0.00%
[system]	0.00	0.00%	0.00	0.00%	8341M	6.49%	9903M	7.46%	-	-
global	0.02	0.04%	-	-	702M	0.54%	803M	0.60%	0	0.00%
zone01	0.00	0.03%	-	-	310M	0.24%	314M	0.23%	0	0.00%
:										

CPU使用量を確認できます。

- USED
使用されているリソース量
- %PART
使用されているリソース割合

物理メモリ使用量 (PhysMem) 、および
仮想メモリ使用量 (VirtMem) を確認できます。

- USED
使用されているリソース量
- %USED
使用されているリソース割合

ネットワーク使用量を確認できます。

- PBYTE
物理ネットワークで使用したバイト数
- %PUSE
使用可能な合計物理帯域幅の割合

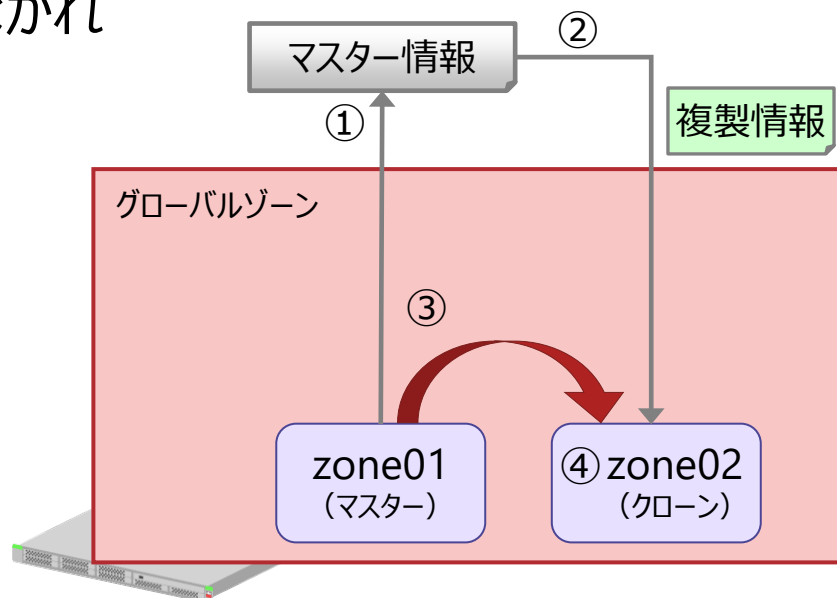


- 停止する場合は、[Ctrl]+[C]キーを押します。
- 上記の例では、5秒間隔で情報が出力されます。

4.ゾーン（仮想Solaris環境）の複製

既存のゾーンの複製（クローン）を作成する方法を説明します。
本番環境のゾーンを複製し、検証環境として使用するゾーンを構築できます。

■ ゾーン複製のながれ



①構成情報（マスター）から複製用の構成情報を作成

複製元ゾーンの構成情報（マスター）を出力し、複製用の構成情報ファイルを作成します。

②構成情報の編集（複製情報の作成）

複製元ゾーンの構成情報（マスター）を元に、新しいゾーンの構成情報（複製）を作成します（IPアドレスとzonepathは重複しないように変更）。

③ゾーンの複製（クローン）の作成

新たな構成情報を元に、ゾーンのクローンを作成します。

④ゾーンの複製（クローン）の初期設定

作成したゾーンにログインし、rootパスワード、ホスト名などを設定します。

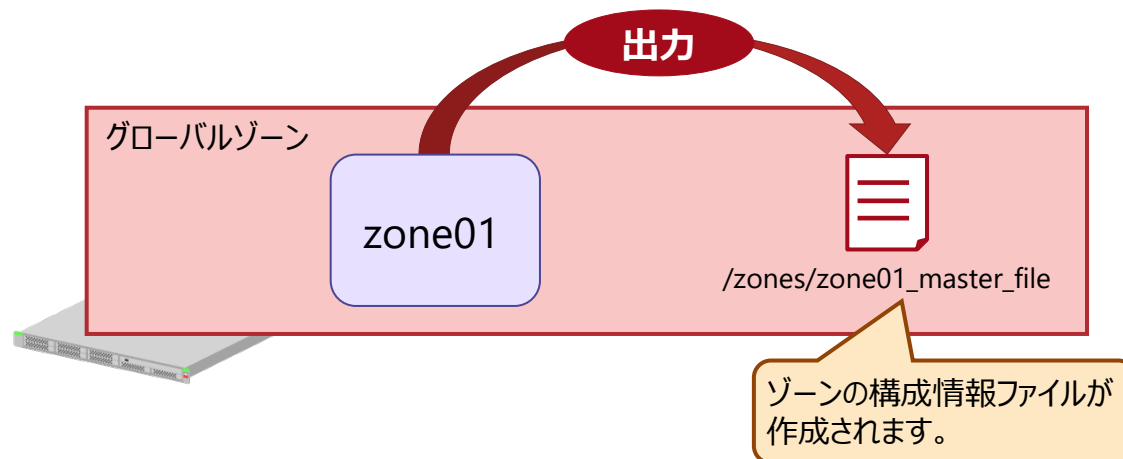


- 本番環境のゾーンを複製することで、検証環境のゾーンを構築することができます。
- 複製先ゾーン（クローン）は、ZFSスナップショット機能とZFSクローン機能によって、短時間で構築されます。

① 構成情報（マスター）から複製用の構成情報を作成

- ✓ ゾーンの構成情報（マスター）の出力（zonecfgコマンド）
【書式】 zonecfg -z <ゾーン名> export -f <構成情報ファイル名>

```
# zonecfg -z zone01 export -f /zones/zone01_master_file
```



② 構成情報の編集（複製情報の作成）

- zonepathなどは複製元の構成と重複するため、変更する必要があります。

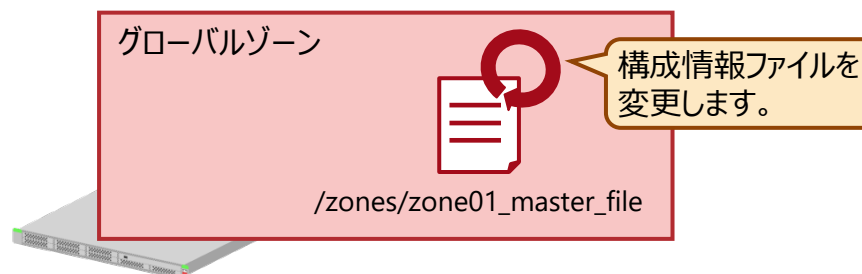
✓ ゾーンの構成情報の編集

※ viエディタを使用した編集例です。

```
# vi /zones/zone01_master_file
```

```
create -b
set brand=solaris
set zonepath=/zones/zones_clone/zone02
set pool=pool_1
set scheduling-class=FSS
add anet
set linkname=vnic0
set lower-link=net0
set configure-allowed-address=true
end
```

zonepathを変更します。



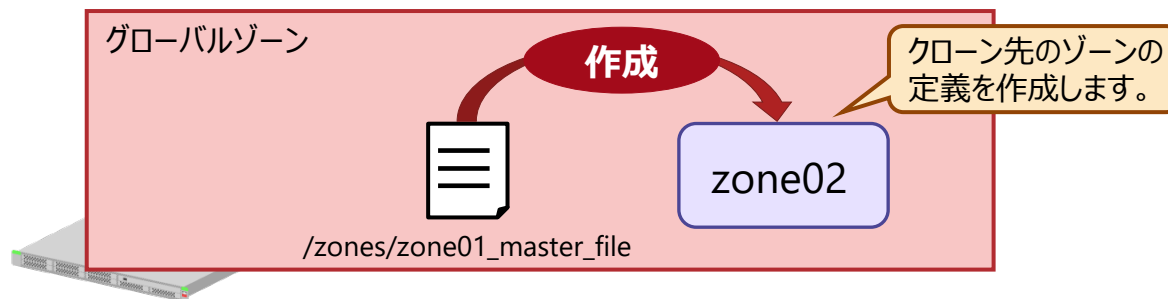
③ ゾーンの複製（クローン）の作成

✓ ゾーン作成（zonecfgコマンド）

【書式】 zonecfg -z <新ゾーン名> -f <構成情報ファイル名>

```
# zonecfg -z zone02 -f /zones/zone01_master_file
```

※この時点では、zone02の定義のみが作成されています。OSやデータは存在しません。

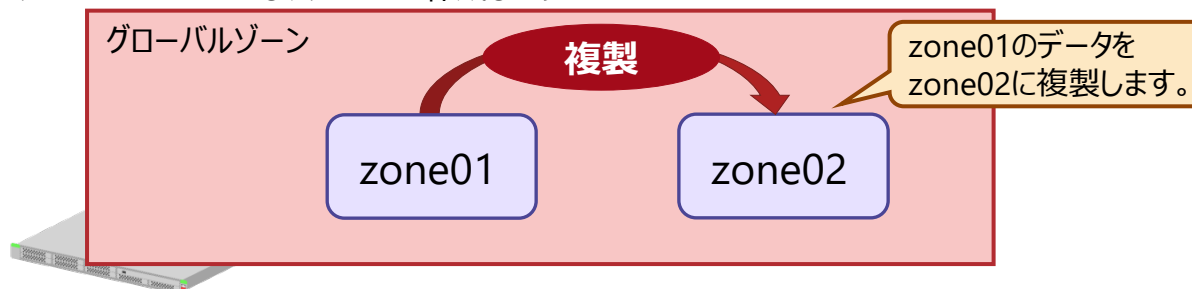


✓ 複製（クローン）の作成（zoneadmコマンド）

【書式】 zoneadm -z <クローン先ゾーン名> clone <クローン元ゾーン名>

```
# zoneadm -z zone02 clone zone01
```

※zone01のデータをzone02にコピーし、クローンを作成します。



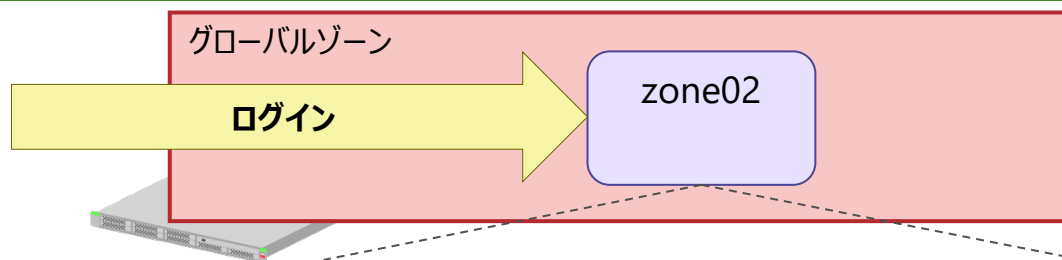
• 1つの環境のクローン作成時間は、数十秒です。

④ ゾーンの複製（クローン）の初期設定

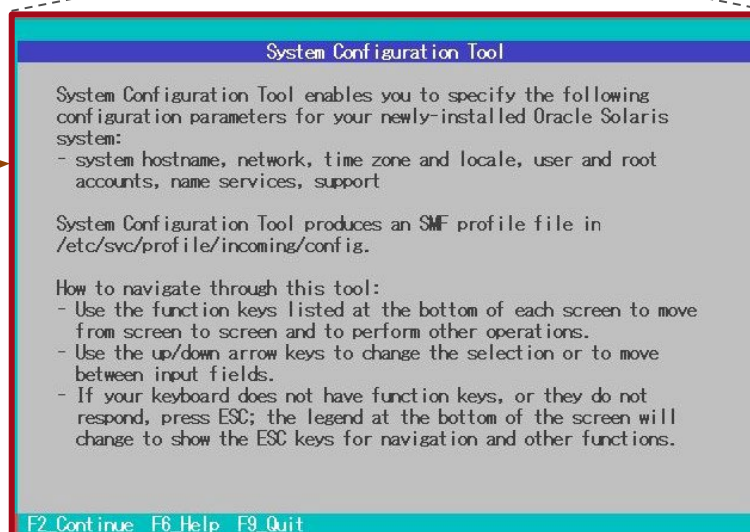
- クローンで作成したゾーンを起動したあと、ゾーンにログインすると、初期設定のツールが起動します。

- ✓ ゾーンへコンソールログイン
【書式】 `zlogin -C <ゾーン名>`

```
# zlogin -C zone02
```



-Cオプションを付けてゾーンへログインすると、初回是对話形式による初期設定画面が表示されます。

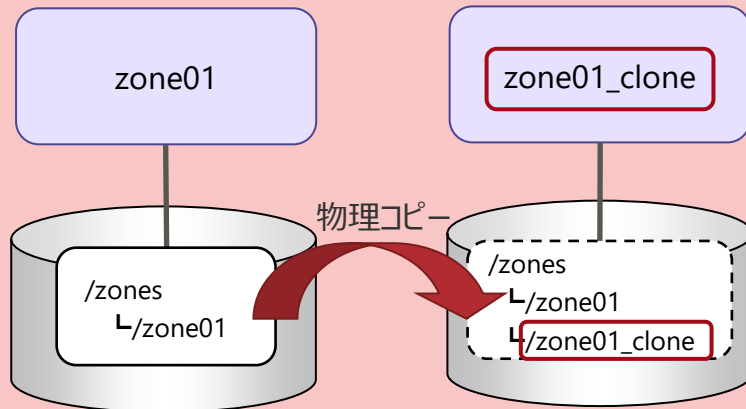


• -Cオプションを付けないと、初期設定画面は表示されません。

《参考》UFSとZFSにおけるゾーン複製方法の違い

UFS環境の場合（Solaris 10のみ）

グローバルゾーン

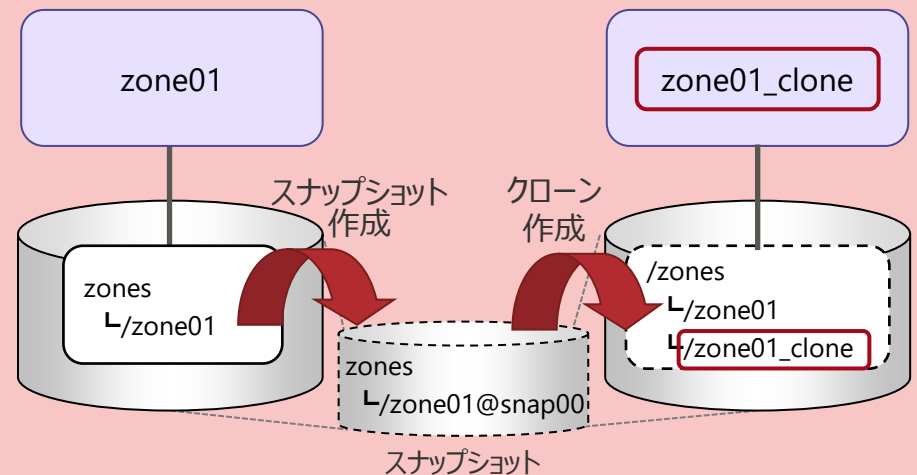


クローンに対して物理的にデータコピーを行っているため、ゾーンの複製には時間を要します。

※5分～10分程度かかります（ディスク使用状況によって増加します）。

ZFS環境の場合

グローバルゾーン



1度スナップショットを作成し、そのスナップショットを元にクローンを作成して、新しいゾーン用の領域としています。データのコピーを実施していないため、ゾーンの複製は短時間（数十秒）で完了します。



- スナップショットとクローンの連携によって、効率的に複数のゾーン環境を構築できます。
- バックグラウンドの動作の違いであるため、UFS環境とZFS環境でユーザーの操作に変更はありません。
- Solaris 11では、ZFS上のみゾーンを作成することができます。

ゾーンの複製（クローン）を作成することで、データベース環境を複製できます。
クローン作成前後の手順のポイントは以下のとおりです。

■ ゾーン構成情報の編集（クローン作成前）

① ゾーン名の変更

複製元と異なる名前を設定します。

② zonepathの変更

複製元のzonepathと異なるパスを設定します。

③ IPアドレスの変更

複製元と異なるIPアドレスを設定します
（ip-type=sharedの場合のみ）。

④ リソース上限設定

CPUとメモリの使用制限する場合に設定します。

⑤ デバイス情報の変更

DB用デバイスを変更します。

■ ミドルウェア用にDB各種設定ファイルを変更（クローン作成後）

① デバイスファイルの削除

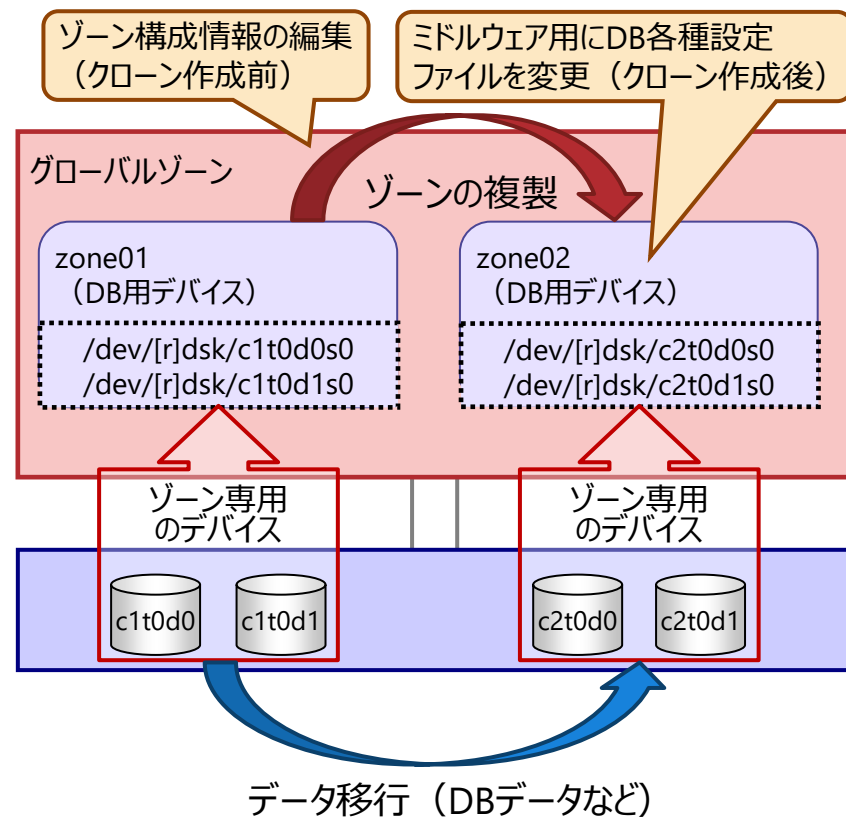
複製元（マスター）のデバイスファイルが残っているので、削除します。

② DBデータの移行

複製元（マスター）で作成したDBデータなど、一式をコピーします（OSコマンドやストレージ機能で実行）。

③ OSのホスト名、IPアドレス変更の影響を受けるデータベース側の設定変更

ホスト名やIPアドレスが記載されたファイルの編集や、コマンドによる構成情報の再作成を実行します。



5.ゾーン（仮想Solaris環境）の移動

既存のゾーンを別のサーバへ移動し、同じゾーン環境を復元する方法を説明します。

ゾーンを移動させることで、災害発生などに備えたバックアップシステムを構築できます。

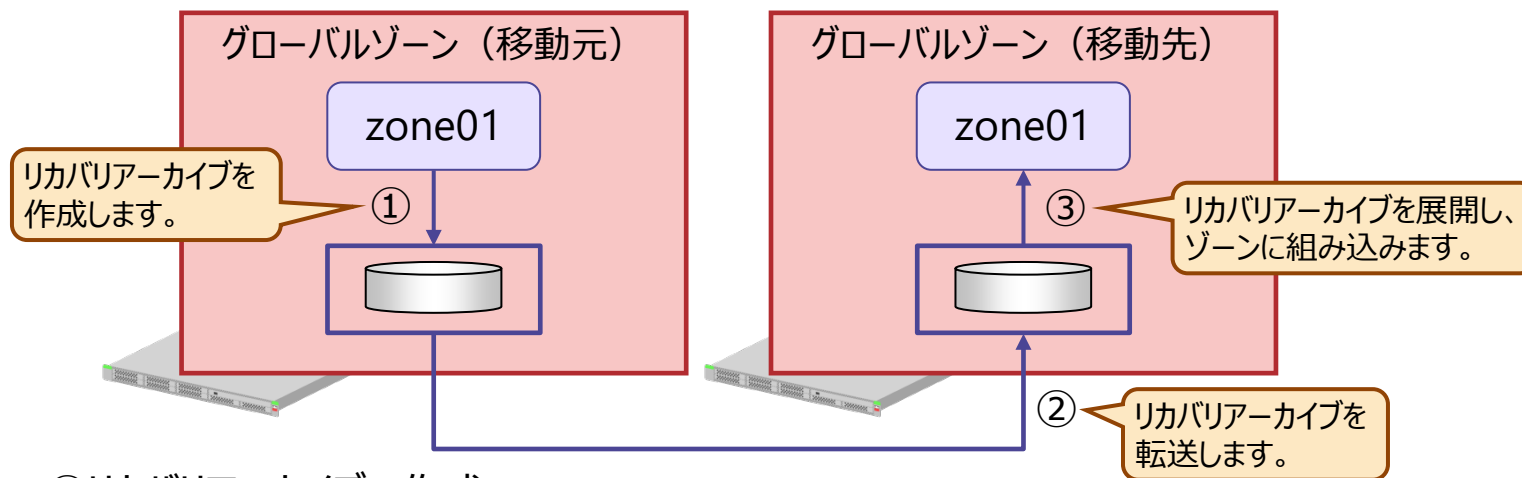
■ ゾーンの移動とは

ゾーンを別のサーバへ移動し、同じゾーン環境を復元することです。

■ ゾーンの移動のながれ

■ 統合アーカイブを使用したゾーンの移動

※ 統合アーカイブを使用したゾーンの移動は、Solaris 11.2 以降の環境が対象となります。Solaris 11.1以前の環境の場合は、「[ゾーン移動の概要 2/2](#)」の「ZFSアーカイブを使用したゾーンの移動」を実施してください。



① リカバリーアーカイブの作成

リカバリーアーカイブを作成します（ゾーンを停止せずに作成可能です）。

② リカバリーアーカイブの転送

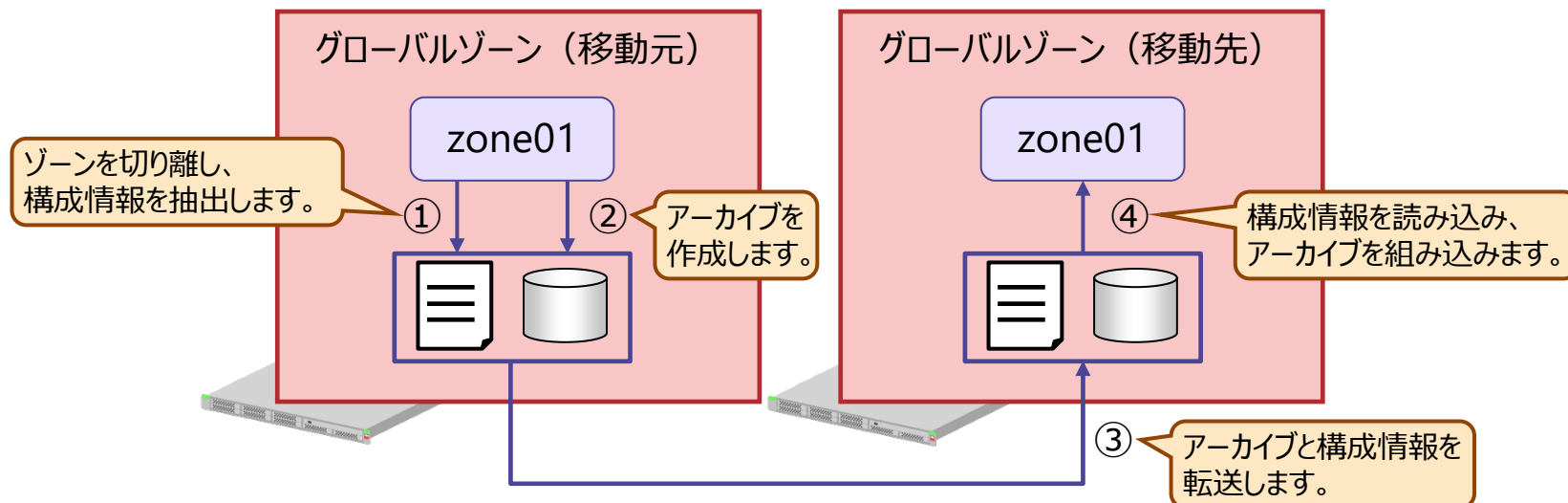
作成したリカバリーアーカイブを別サーバへ転送します。転送完了後、移動させるゾーンを停止します。

③ リカバリーアーカイブの展開とゾーンの組み込み

別サーバでリカバリーアーカイブを展開し、ゾーンに組み込みます。

■ ZFSアーカイブを使用したゾーンの移動

※ ZFSアーカイブを使用したゾーンの移動は、Solaris 11.3以前の環境が対象となります。Solaris 11.4の環境の場合は、「[ゾーン移動の概要 1/2](#)」の「統合アーカイブを使用したゾーンの移動」を実施してください。



①ゾーンの切り離しと構成情報の抽出

移動させるゾーンを停止しておき、グローバルゾーンから切り離し、構成情報を抽出します。

②アーカイブの作成

アーカイブを作成します。

③アーカイブの転送

作成したアーカイブを構成情報とあわせて別サーバへ転送します。

④アーカイブの展開とゾーンの組み込み

別サーバでアーカイブを展開し、構成情報とアーカイブを組み込みます。



・本番環境のゾーンを移動することで、災害発生時に利用できるバックアップシステム（災対環境）を構築できます。

- ✓ ザーンのリカバリーアーカイブを作成

【書式】 archiveadm create -r -z <ゾーン名> <リカバリーアーカイブ名.uar>

```
# archiveadm create -r -z zone01 /zones/zone01-archive.uar
```

- ✓ アーカイブを転送

作成したリカバリーアーカイブを、FTPやNFSで移動先のサーバに転送します。

- ✓ ザーンの停止

【書式】 zoneadm -z <ゾーン名> shutdown

```
# zoneadm -z zone01 shutdown
```

③ リカバリアーカイブの展開とゾーンの組み込み（移動先での操作）

✓ ゾーンの構成

【書式】 zonecfg -z <ゾーン名> create -a <リカバリアーカイブファイル名>

```
# zonecfg -z zone01 create -a /export/home/user01/zone01-archive.uar
```

✓ ゾーンの組み込み

【書式】 zoneadm -z <ゾーン名> install -a <リカバリアーカイブファイル名>

```
# zoneadm -z zone01 install -a /export/home/user01/zone01-archive.uar
```

✓ ゾーンの起動

【書式】 zoneadm -z <ゾーン名> boot

```
# zoneadm -z zone01 boot
```



- 移動元のSolaris 11環境にインストールされているパッケージが、移動先のSolaris 11環境のバージョンと比べて古い場合、installコマンドに-Uオプションを使用してパッケージを同期します。

■ ZFSアーカイブを使用したゾーンの移動

① ゾーンの切り離しと構成情報の抽出（移動元での操作）

- ✓ ゾーンの切り離し（detach）

【書式】 zoneadm -z <ゾーン名> detach

```
# zoneadm -z zone01 detach
```

- ✓ ゾーンの構成情報の抽出

【書式】 zonecfg -z <ゾーン名> export -f <構成情報ファイル名>

```
# zonecfg -z zone01 export -f /zones/zone01.zonecfg
```

② アーカイブの作成（移動元での操作）

- ✓ ゾーンのファイルシステム（zonepath）のスナップショットの作成

【書式】 zfs snapshot -r <ファイルシステム>@<スナップショット名>

```
# zfs snapshot -r zones/zone01@v2v
```

- ✓ スナップショットからアーカイブを作成

【書式】 zfs send [オプション] <ファイルシステム>@<スナップショット名>

```
# zfs send -rc zones/zone01@v2v | gzip > /zones/zone01.zfs.gz
```

③ アーカイブの転送（移動元での操作）

- ✓ アーカイブを転送

作成したアーカイブと構成情報（ゾーン設定ファイル）を、FTPやNFSで移動先のサーバに転送します。

④ アーカイブの展開とゾーンの組み込み（移動先での操作）

✓ ゾーンの構成情報を読み込み

【書式】 zonecfg -z <ゾーン名> -f <構成情報ファイル名>

```
# zonecfg -z zone01 -f /export/zone01.zonecfg
```

✓ ゾーンの組み込み

【書式】 zoneadm -z <ゾーン名> install -p -a <アーカイブファイル名>

```
# zoneadm -z zone01 install -p -a /export/zone01.zfs.gz
```

✓ ゾーンの起動

【書式】 zoneadm -z <ゾーン名> boot

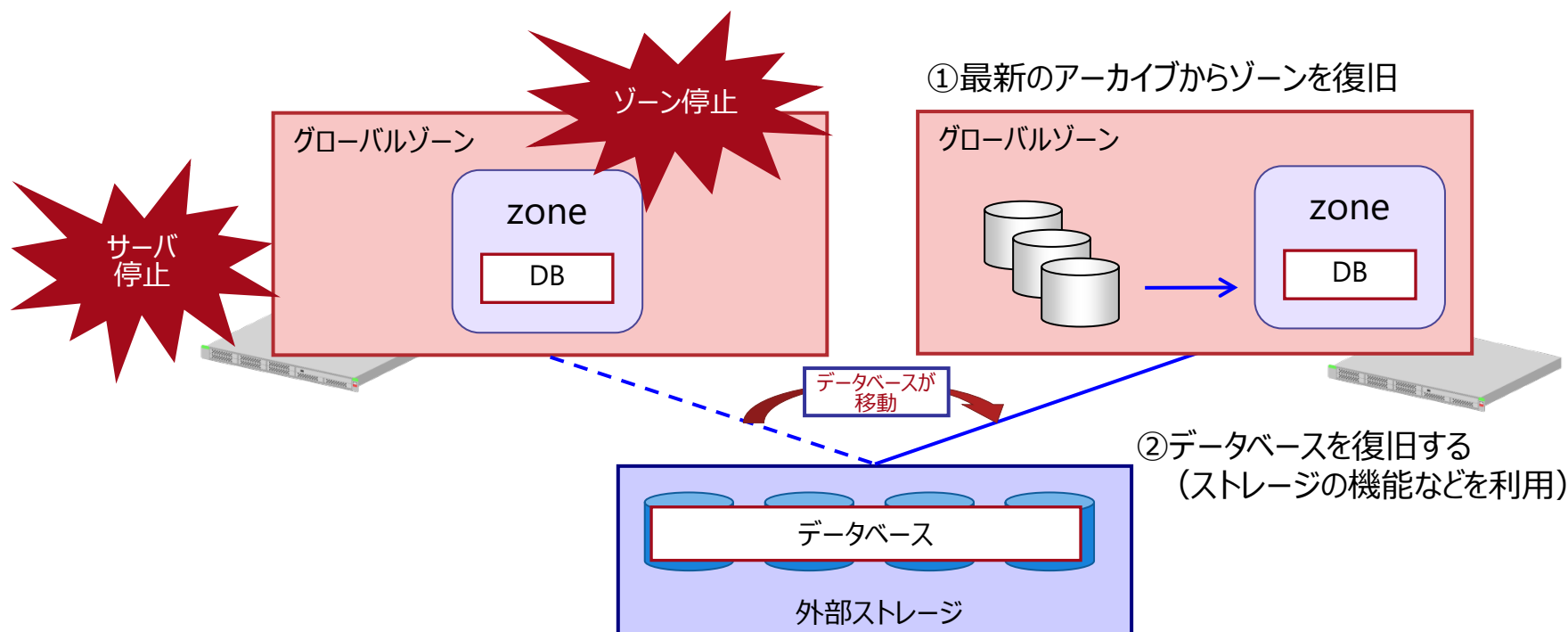
```
# zoneadm -z zone01 boot
```



• 移動元のSolaris 11環境にインストールされているパッケージが、移動先のSolaris 11環境のバージョンと比べて古い場合、installコマンドに-Uオプションを使用してパッケージを同期します。

退避済みのアーカイブデータを利用し、迅速にデータベース環境を復旧できます。

- 運用では、データベース環境の更新（アップデート適用など）時に最新のアーカイブを災対環境へ退避します。
- 保守時や障害時は、別サーバ上でゾーンを復元し、代替環境として業務を継続します。



- ゾーンの切り替えによって、外部ストレージ上のデータの不整合を生じさせないようにする必要があります。本書では、サーバ間のストレージ共有構成や、ETERNUSのREC（Remote Equivalent Copy）機能によるストレージ間のデータ同期などの導入を想定しています。

6.ゾーン（仮想Solaris環境）の CPUリソース操作

ゾーンが使用するCPUリソースの配分比率や上限値を設定したり、ゾーン間でCPUリソースを動的に移動させたりする操作を説明します。

■ ゾーンのCPUリソース操作と使用するコマンド

■ CPUシェア数の設定（prctlコマンド）

- 各ゾーンが使用するCPUリソースの配分比率を設定します。

■ CPUリソースのキャッピング設定（zonecfgコマンド、prctlコマンド）

- 各ゾーンが使用するCPUリソースの上限値を設定します。

■ CPUリソースの動的移動（poolcfgコマンド、pooldデーモン）

- ゾーンを起動したまま、リソース負荷の状況に応じて、設定範囲内でCPUリソースを自動的に移動させます。

■ 各ゾーンが使用するCPUリソースの配分比率を設定します。

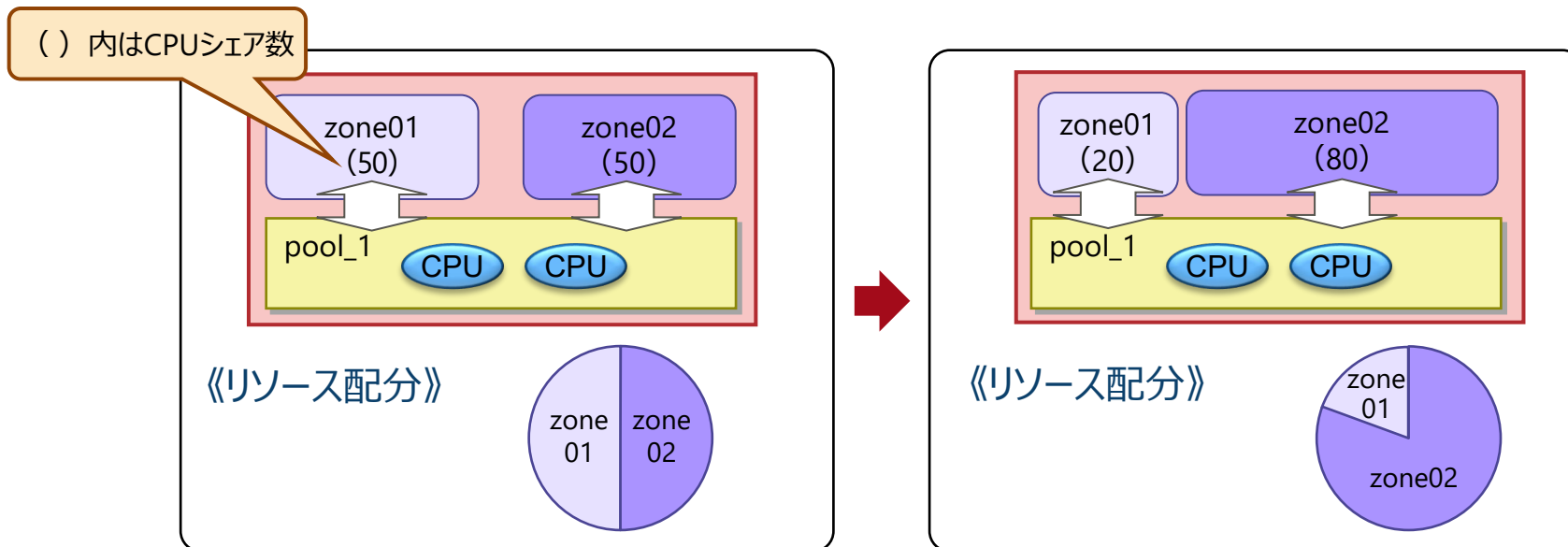
※CPUシェア数の設定によるCPUリソース配分は、リソースプールのスケジューラが「FSS」の場合のみ有効です。

✓ CPUシェア数の設定（prctlコマンド）

【書式】 prctl [-n <リソース制御名>] [-r] [-v <値>] [-i <IDタイプ>] <ゾーン名>

```
# prctl -n zone.cpu-shares -r -v 20 -i zone zone01  
# prctl -n zone.cpu-shares -r -v 80 -i zone zone02
```

zone01に20 %、zone2に80 %の比率でCPUリソースを配分します。



■ 各ゾーンが使用するCPUリソースの上限値を設定します。

✓ CPUキャップ設定（capped-cpuプロパティ／zonecfgコマンド）

CPU数を指定してゾーンの使用上限値を設定します。

設定値には、0.75や1.25など小数を使用できます（0.75は75 %、1.25は125 %を意味します）。

【書式】 zonecfg -z <ゾーン名>

```
# zonecfg -z zone02
zonecfg:zone02>add capped-cpu
zonecfg:zone02:capped-cpu>set ncpus=1
zonecfg:zone02:capped-cpu>end
```

ncpusの値に、zone02が占有するCPU数を指定します。
ncpus=1は、1つのCPUの100 %を意味し、1つのCPUを占有します。

✓ 一時的なCPUキャップ設定（zone.cpu-capリソース制御／prctlコマンド）

zonecfgコマンドで設定したCPUリソース使用上限値を、ゾーンを起動したまま一時的に変更します。

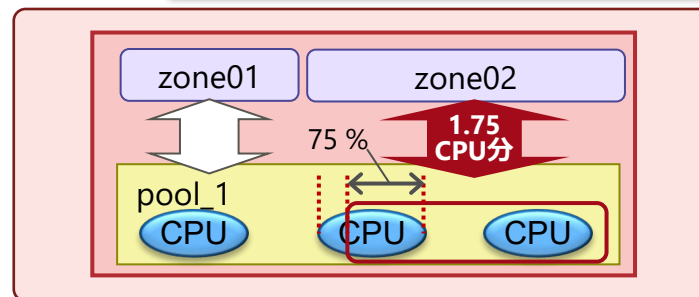
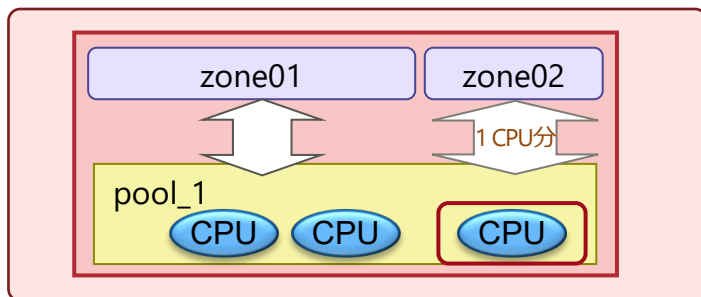
変更内容は、ゾーンを再起動すると元に戻ります。

設定値には整数を使用します（75は75 %、125は125 %を意味します）。

【書式】 prctl [-n <リソース制御名>] [-r] [-v <値>] [-i <IDタイプ>] <ゾーン名>

```
# prctl -n zone.cpu-cap -r -v 175 -i zone zone02
```

zone.cpu-capパラメーターに175を指定し、CPU1.75個分まで占有できるよう変更します。



- capped-cpu プロパティの設定値は、zone.cpu-cap リソース制御の100分の1になります。設定値1は、zone.cpu-cap リソース制御の設定値100に等しくなります。

- ゾーンを起動したまま、リソース負荷の状況に応じて、設定範囲内でCPUリソースを移動させます。

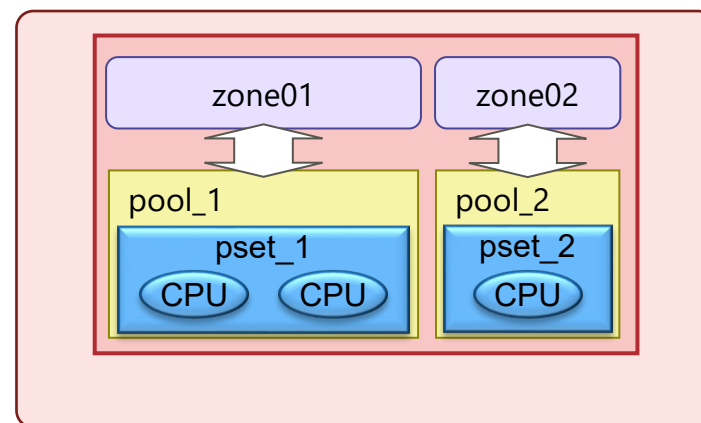
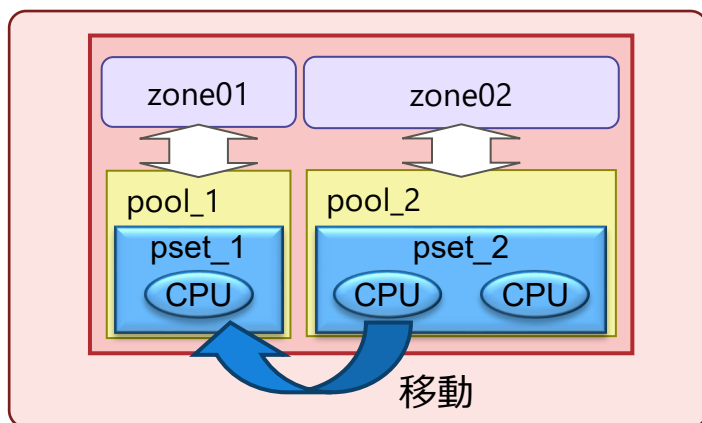
- 手動と自動の2つの方法があります。

- ✓ 手動でのCPUリソースの移動（poolcfgコマンド）

【書式】 poolcfg [-d] [-c <コマンド>]

```
# poolcfg -dc 'transfer 1 from pset pset_2 to pset_1'
```

pset_2からpset_1へ1 CPU移動させます。



- ✓ 自動でのCPUリソースの移動（pooldデーモン）

【書式】 svcadm enable <サービス名>

```
# svcadm enable svc:/system/pools/dynamic:default
```

自動でのCPUリソースの移動を有効にします。



- pooldデーモンによる自動CPUリソース移動は、サービス起動時に即有効になります。リソースプールの負荷状況に応じて、割り当てられるCPU数の範囲内で、CPUを自動的に移動させます。
→リソースプールに割り当てられるCPU数の範囲については、「[リソースプール作成の概要 2/2](#)」を参照してください。

7.ゾーン（仮想Solaris環境）の 仮想ネットワークの応用

ゾーンを利用した仮想ネットワーク機能の応用として、以下の機能を解説します。

- etherstub
- IPMP
- 帯域幅制限
- VLAN

■ ネットワークサービスの確認

- ネットワークサービス（svc:/network/physical:default）の起動状態を確認します。

※ Solaris 11/11で仮想ネットワークを構築する場合は、svc:/network/physical:defaultサービスが起動していること、および svc:/network/physical:nwamサービスが停止していることを確認します。

- ✓ svc:/network/physical:default（パッケージ）のネットワークサービスの確認

【書式】svcs <サービス名>

```
# svcs svc:/network/physical:default
STATE          STIME          FMRI
online         18:52:41      svc:/network/physical:default
```

svc:/network/physical:default は、システムの基本的なネットワーク設定を**手動**で行うサービスです。

- ✓ nwamサービスの確認（Oracle Solaris 11 11/11の場合のみ）

【書式】svcs <サービス名>

```
# svcs svc:/network/physical:nwam
STATE          STIME          FMRI
disabled       Sep_20         svc:/network/physical:nwam
```

svc:/network/physical:nwamは、システムの基本的なネットワーク設定を**自動**で行うサービスです。

■ etherstubの設定

- etherstubとは、グローバルゾーン上に作成できる仮想スイッチ機能です。
- ゾーン間におけるサーバ内部のネットワークを構築できます。
 - 物理ネットワークよりも、高いスループット／レスポンス性能の環境を構築できます。
 - サーバ外部を経由しない通信のため、高いセキュリティを保つことができます。
- etherstubを作成し、プライベート仮想ネットワークを構築します。

✓ etherstubの作成

【書式】 `dladm create-etherstub <etherstub名>`

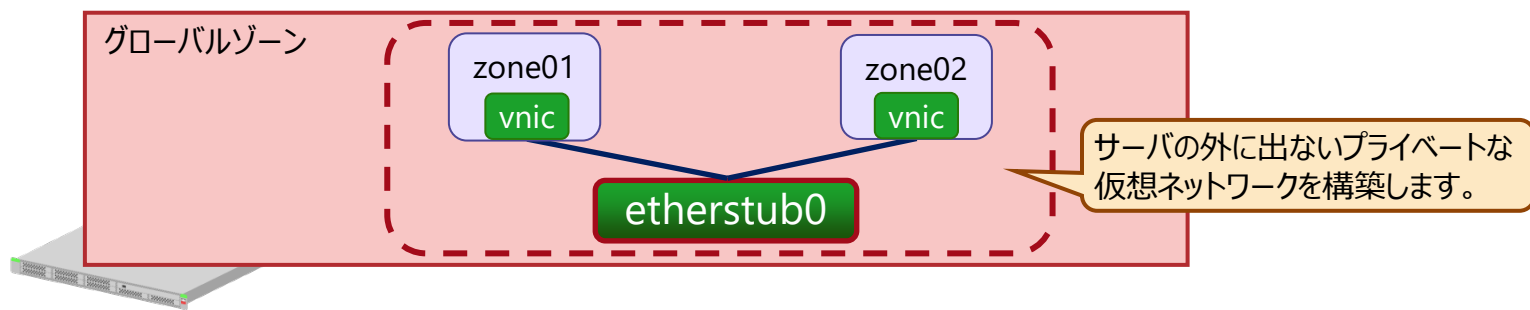
```
# dladm create-etherstub etherstub0
```

✓ ゾーンにetherstubを割り当て

【書式】 `zonecfg -z <ゾーン名>`

```
zonecfg:zone01> select anet lower-link=vnic0  
zonecfg:zone01:anet> set lower-link=etherstub0
```

作成したetherstubをlower-linkに指定します。
zone02も同様に設定します。



- 💡
- etherstubで接続したゾーン同士のみが通信可能な仮想ネットワークを構築できます。グローバルゾーンからも通信することができないため、完全に閉じたネットワークとなります。

■ IPMPの設定

- IPMPを作成し、ゾーンのネットワークを冗長化します。

✓ IPMPの作成

【書式】 ipadm create-ipmp <IPMP名>

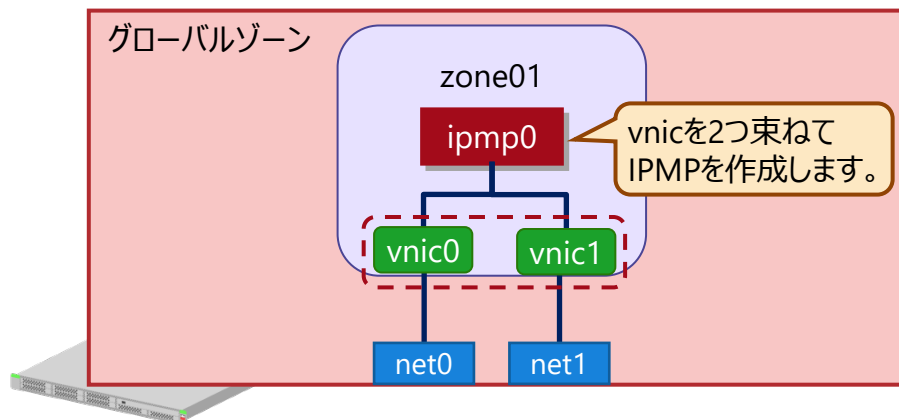
```
zone01# ipadm create-ipmp ipmp0
```

✓ IPMPへvnicを組み込み

【書式】 ipadm add-ipmp -i <vnic名> [-i <VNIC名>] <IPMP名>

```
zone01# ipadm add-ipmp -i vnic0 -i vnic1 ipmp0
```

ゾーンに割り当てられている2つのvnicをIPMPに組み込みます。



- 本手順でIPMPを作成した場合、「分散型構成」で「リンクベース」となります。詳細については、Oracle社のマニュアルを参照してください。

https://docs.oracle.com/cd/E37838_01/html/E60991/gfkcy.html#scrolltoc

■ 帯域幅制限の設定

- Solaris 11では、物理NICまたは仮想NICに対して帯域幅制限を設定できます。

✓ フローの作成 (flowadmコマンド)

【書式】 flowadm add-flow -l <リンク名> -a local_ip=<IPアドレス> <フロー名>

```
# flowadm add-flow -l vnic0 -a local_ip=192.168.10.XX vnic0_flow
```

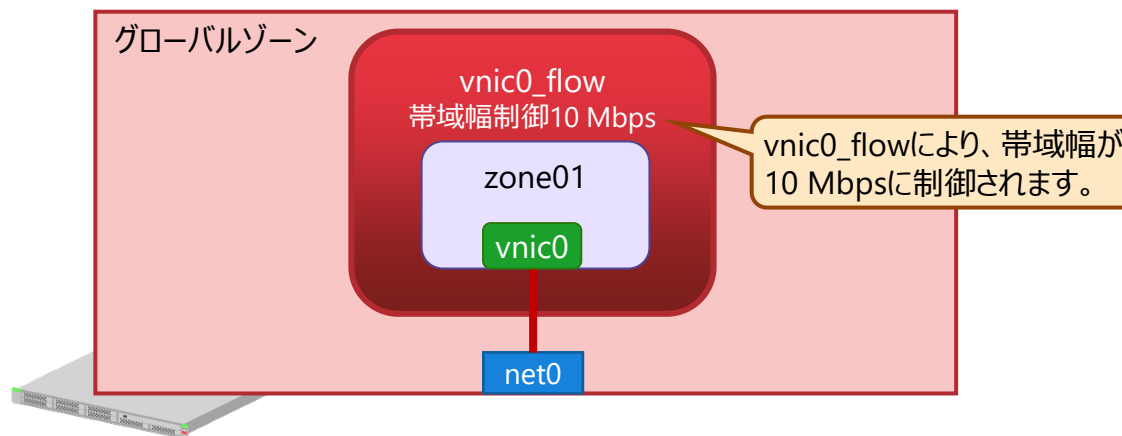
vnic0に対して、
フロー-vnic0_flowを作成します。

✓ フローのプロパティ (帯域幅制限) の設定 (flowadmコマンド)

【書式】 flowadm set-flowprop -p max-bw=<値> <フロー名>

```
# flowadm set-flowprop -p max-bw=10M vnic0_flow
```

vnic0_flowを通して、vnic0に
10 Mbpsの帯域幅制限を設定します。



- 設定した帯域幅制限は、容量の大きいファイルをダウンロードして確認できます。
→詳しくは、「[《参考》帯域幅制限の確認方法](#)」を参照してください。

■ VLANの設定

- VLANを作成することで、物理NICを複数のネットワークに分割できます。
本書では、exclusive方式のnetリソースを使用する設定例を記載しています。

✓ VLANの作成

【書式】 `dladm create-vlan -l <リンク名> -v <VID> <vlan名>`

```
# dladm create-vlan -l net0 -v 123 vlan0
# dladm create-vlan -l net0 -v 456 vlan1
```

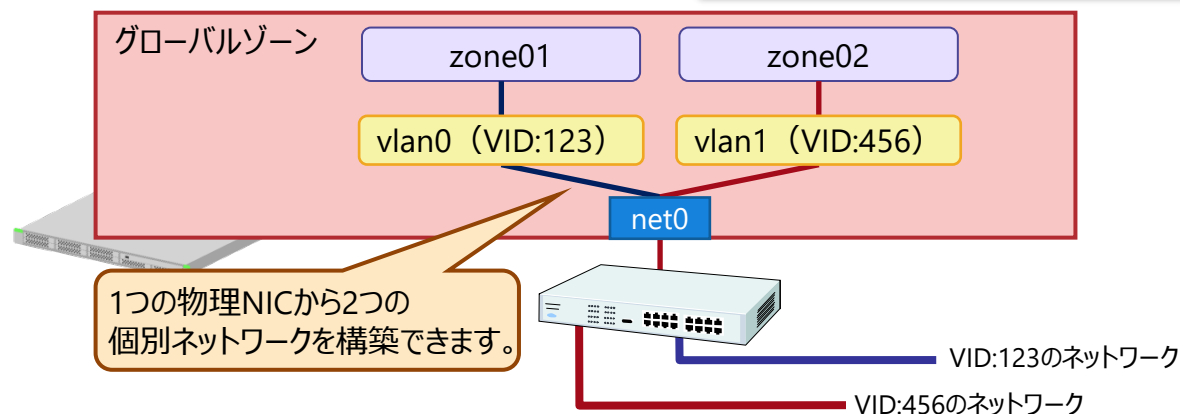
VLANを2個作成します。

✓ ゾーンにVLANを割り当て

【書式】 `zonecfg -z <ゾーン名>`

```
# zonecfg -z zone01
zonecfg:zone01> add net
zonecfg:zone01:net> set physical=vlan0
```

作成したvlan0をzone01のphysicalに指定します。
zone02も同様の手順でvlan1を割り当てます。



• 作成したVLANのVIDをスイッチ側で設定する必要があります。

《参考》帯域幅制限の確認方法

容量の大きいファイルをダウンロードすることで、帯域幅制限を確認できます。

✓ 帯域幅制限の確認（flowstatコマンド）

ゾーンに対して帯域幅制限を設定した状態で、グローバルゾーンから容量の大きいファイルをダウンロードします。
ファイルダウンロード中に、以下のコマンドで帯域幅制限の動作を確認できます。

【書式】flowstat <フロー名> [interval(秒) [回数]]

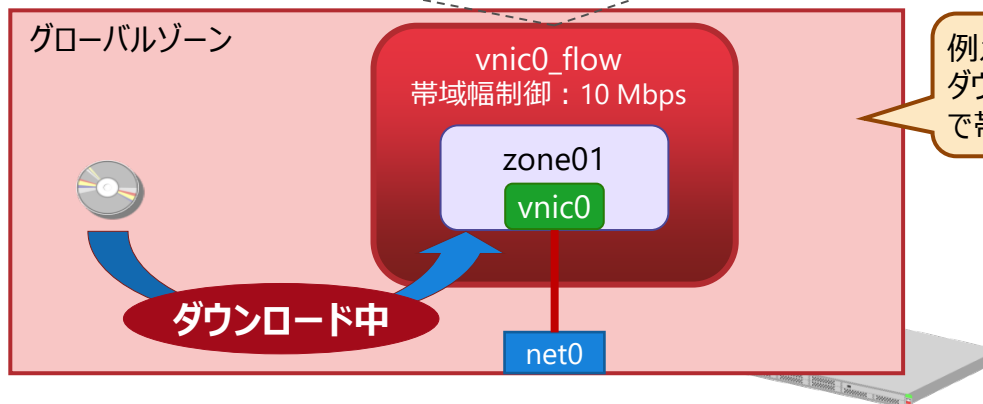
```
# flowstat -i 1 vnic0_flow
```

FLOW	IPKTS	RBYTES	IDROPS	OPKTS	OBYTES	ODROPS
vnic0_flow	0	0	0	0	0	0
vnic0_flow	939	1.27M	70	483	32.25K	0
vnic0_flow	893	1.21M	66	469	31.70K	0
vnic0_flow	899	1.25M	72	472	31.62K	0
vnic0_flow	908	1.21M	76	479	32.26K	0
:						

※ flowstatのRBYTESの表示がMB/s単位のため、10 Mbpsで帯域制御した場合は、約1.25 MB/sと出力されます。

受信パケット

送信パケット



付録

『Manually Installing an Oracle Solaris 11.4 System』 (Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E69250.pdf

『Automatically Installing Oracle Solaris 11.4 Systems』 (Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E60976.pdf

『Creating Package Repositories in Oracle Solaris 11.4』 (Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E60982.pdf

『Creating and Administering Oracle Solaris 11.4 Boot Environments』 (Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E60980.pdf

『Oracle Solaris ゾーンの作成と使用』 (Oracle社)

https://docs.oracle.com/cd/E75431_01/pdf/E75320.pdf

『Oracle Solaris Zones Configuration Resources』 (Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E61040.pdf

『Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4』
(Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E60991.pdf

『Managing Network Virtualization and Network Resources in Oracle Solaris 11.4』
(Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E60989.pdf

『Managing Network Datalinks in Oracle Solaris 11.4』 (Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E60990.pdf

SPARCサーバ／Oracle Solarisの技術情報を掲載



今すぐアクセス!!!

<https://www.fujitsu.com/jp/sparc-technical/>

版数	更新日時	更新内容
初版	2012年7月	新規作成
第2.0版	2013年6月	Oracle Solaris 11.1 に対応
第2.1版	2014年5月	誤記訂正、デザインの変更
第4.0版	2019年6月	Oracle Solaris 11.4 に対応

■ 使用条件

■ 著作権・商標権・その他の知的財産権について

- コンテンツ（文書・画像・音声等）は、著作権・商標権・その他の知的財産権で保護されています。本コンテンツは、個人的に使用する範囲でプリントアウトまたはダウンロードできます。ただし、これ以外の利用（ご自分のページへの再利用や他のサーバへのアップロード等）については、当社または権利者の許諾が必要となります。

■ 保証の制限

- 本コンテンツについて、当社は、その正確性、商品性、ご利用目的への適合性等に関して保証するものではなく、そのご利用により生じた損害について、当社は法律上のいかなる責任も負いかねます。本コンテンツは、予告なく変更・廃止されることがあります。

■ 輸出または提供

- 本製品を輸出又は提供する場合は、外国為替及び外国貿易法及び米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとり下さい。

■ 商標

- UNIXは、米国およびその他の国におけるオープン・グループの登録商標です。
- SPARC Enterprise、SPARC64、SPARC64 ロゴおよびすべてのSPARC商標は、米国SPARC International, Inc.のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
- OracleとJavaは、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- その他各種製品名は、各社の製品名称、商標または登録商標です。

