

在 Oracle® Solaris 11.2 中管理网络虚拟化和 网络资源

ORACLE®

文件号码 E53793-02
2014 年 9 月

版权所有 © 2011, 2014, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

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.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

目录

使用本文档	9
1 网络虚拟化和网络资源管理简介	11
Oracle Solaris 11.2 中在管理网络虚拟化和网络资源方面的新增功能	11
什么是网络虚拟化和网络资源管理？	12
虚拟网络概述	13
虚拟网络组件	13
虚拟网络的工作方式	15
使用虚拟可扩展局域网	17
使用边缘虚拟桥接	17
谁应实现虚拟网络？	18
网络资源管理概述	18
通过使用数据链路属性进行网络资源管理	19
通过使用流管理网络资源	19
网络资源管理的益处	19
2 创建和管理虚拟网络	21
配置虚拟网络的组件	21
用于配置虚拟网络组件的命令	21
▼ 如何配置 VNIC 和 Etherstub	22
▼ 如何配置带有 VLAN ID 的 VNIC	24
构建虚拟网络	25
▼ 如何为虚拟网络配置区域	26
▼ 如何重新配置区域以使用 VNIC	28
▼ 如何临时在区域中创建 VNIC	30
▼ 如何配置专用虚拟网络	31
管理 VNIC	34
显示 VNIC	34
修改 VNIC 的 VLAN ID	38
修改 VNIC MAC 地址	39

迁移 VNIC	41
删除 VNIC	42
将单根 I/O 虚拟化与 VNIC 一起使用	45
启用数据链路的 SR-IOV 模式	45
创建 VF VNIC	46
迁移 VF VNIC	47
显示 VF 信息	48
3 使用虚拟可扩展局域网配置虚拟网络	51
VXLAN 概述	51
使用 VXLAN 的优点	52
VXLAN 命名约定	52
VXLAN 拓扑	53
结合使用 VXLAN 和区域	55
规划 VXLAN 配置	57
VXLAN 要求	57
配置 VXLAN	57
▼ 如何配置 VXLAN	57
显示 VXLAN 信息	61
删除 VXLAN	62
将 VXLAN 分配给区域	62
▼ 如何将 VXLAN 分配给区域	62
使用案例：在链路聚合上配置 VXLAN	63
4 使用边缘虚拟桥接管理服务器-网络边缘虚拟化	67
服务器-网络边缘虚拟化中的 EVB 支持	67
反射中继	68
在网络中自动进行 VNIC 配置	68
通过使用 EVB 改进网络和服务器效率	68
安装 EVB	71
▼ 如何安装 EVB	71
控制基于同一物理端口的 VM 之间的切换	71
允许 VM 通过外部交换机通信	72
使用 LLDP 管理 VM 之间的通信	76
使用 VDP 交换 VNIC 信息	77
VDP 如何交换 VNIC 信息	78
显示 VDP 和 ECP 状态和统计数据	79
显示 VDP 状态和统计数据	79
显示链路属性	79

显示 ECP 状态和统计数据	80
更改缺省 EVB 配置	80
▼ 如何更改缺省 EVB 配置	81
5 关于弹性虚拟交换机	85
弹性虚拟交换机 (Elastic Virtual Switch, EVS) 功能概述	85
Oracle Solaris 中的虚拟交换机	85
Oracle Solaris 弹性虚拟交换机的功能是什么?	87
使用 EVS 的益处	88
弹性虚拟交换机资源	89
EVS 中的名称空间管理	90
EVS 组件	90
EVS 管理器	92
EVS 控制器	92
EVS 客户机	94
EVS 节点	94
EVS 管理命令	94
evsadm 命令	94
evsstat 命令	96
dladm 命令	97
zonecfg 命令	97
有关管理连接到弹性虚拟交换机的 VNIC 的限制	98
自动生成的 VXLAN 数据链路	98
使用 EVS 时必需的软件包	98
EVS 如何使用区域	99
使用 EVS 时的安全要求	99
6 管理弹性虚拟交换机	101
EVS 管理任务	101
规划弹性虚拟交换机配置	102
创建和管理 EVS 控制器	103
EVS 控制器的必需软件包	103
用于配置 EVS 控制器的命令	104
配置 EVS 控制器	105
配置弹性虚拟交换机	112
弹性虚拟交换机的必需软件包	112
用于配置弹性虚拟交换机的命令	112
▼ 如何配置弹性虚拟交换机	114
为弹性虚拟交换机创建 VNIC	115

管理弹性虚拟交换机、IPnet 和 VPort	117
管理弹性虚拟交换机	118
管理 IPnet 配置	121
管理 VPort 配置	123
删除弹性虚拟交换机	127
监视弹性虚拟交换机	128
弹性虚拟交换机的使用案例示例	130
使用案例：配置弹性虚拟交换机	130
使用案例：为租户配置弹性虚拟交换机	136
7 管理网络资源	143
使用数据链路属性管理网络资源	143
用于在数据链路中分配资源的命令	144
管理 NIC 环	144
在 MAC 客户机中分配环	144
在 VLAN 中分配环	145
用于配置环的命令	145
显示数据链路上的环使用和环分配	146
配置客户机并分配环	147
管理池和 CPU	152
使用池和 CPU	152
为数据链路配置 CPU 池	154
向数据链路分配 CPU	156
使用流管理网络资源	157
用于流中的资源分配的命令	158
配置流	158
使用案例：通过设置数据链路和流属性来管理网络资源	160
8 监视网络通信流量和资源使用情况	165
监视数据链路和流的网络通信统计数据的概述	165
用于监视网络通信统计数据的命令	167
显示链路的网络通信统计数据	167
显示网络设备的网络通信统计数据	168
显示数据链路的网络通信统计数据	170
显示链路聚合的网络通信统计数据	171
显示网桥的网络通信统计数据	172
显示流的网络通信统计数据	172
为网络通信配置网络记帐	175
▼ 如何设置网络记账	175

显示网络通信的历史统计数据	176
索引	181

使用本文档

- 概述 - 介绍了如何配置 Oracle Solaris 虚拟网络功能和监视网络通信。它还介绍了用于管理网络资源的不同进程。
- 目标读者 - 系统管理员。
- 必备知识 - 基本和一些高级的网络管理技能。

产品文档库

有关本产品的最新信息和已知问题均包含在文档库中，网址为：<http://www.oracle.com/pls/topic/lookup?ctx=E36784>。

获得 Oracle 支持

Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

反馈

可以在 <http://www.oracle.com/goto/docfeedback> 上提供有关本文档的反馈。

网络虚拟化和网络资源管理简介

本章概述了 Oracle Solaris 中的网络虚拟化和网络资源管理。

本章包含以下主题：

- “Oracle Solaris 11.2 中在管理网络虚拟化和网络资源方面的新增功能” [11]
- “什么是网络虚拟化和网络资源管理？” [12]
- “虚拟网络概述” [13]
- “网络资源管理概述” [18]

Oracle Solaris 11.2 中在管理网络虚拟化和网络资源方面的新增功能

对于现有客户，本节重点介绍了此发行版中的关键更改：

- Oracle Solaris 弹性虚拟交换机 (Elastic Virtual Switch, EVS) 功能 – Oracle Solaris 网络虚拟化功能进行了扩展，现在可以直接管理虚拟交换机。Oracle Solaris 弹性虚拟交换机功能在数据中心或多租户云环境内提供了虚拟网络基础结构，以便互连驻留在多个服务器上的虚拟机。连接到同一个弹性虚拟交换机的虚拟机可以相互通信。通过 EVS 可以集中管理多个主机上的虚拟机以及连接到弹性虚拟交换机的 VNIC。有关更多信息，请参见第 5 章 [关于弹性虚拟交换机](#)。有关如何管理弹性虚拟交换机的更多信息，请参见第 6 章 [管理弹性虚拟交换机](#)。
- 虚拟可扩展局域网 (Virtual Extensible Local Area Network, VXLAN) 支持 – Oracle Solaris 支持 VXLAN 技术，此技术提供了隔离方法以支持大型数据中心的虚拟化。使用此技术可以在属于云环境中不同第 2 层网络的物理服务器之间迁移虚拟机。有关更多信息，请参见第 3 章 [使用虚拟可扩展局域网配置虚拟网络](#)。
- 单根 I/O 虚拟化 (Single Root I/O Virtualization, SR-IOV) 支持 – 使用此功能可以在支持 SR-IOV 的网络设备上创建基于虚拟功能 (virtual function, VF) 的 VNIC。有关更多信息，请参见[“将单根 I/O 虚拟化与 VNIC 一起使用”](#) [45]。
- 在区域中临时创建虚拟网络接口卡 (virtual network interface cards, VNIC) – 可以从全局区域直接在非全局区域中创建临时 VNIC。必须结合使用 `-t` 选项和 `dladm create-vnic` 命令才能创建临时 VNIC。临时 VNIC 将持续到下一次重新引导区域之前。除临时创建 VNIC 之外，您还可以在区域中临时创建 VLAN 和基于 InfiniBand

的 IP (IP over InfiniBand, IPoIB) 分区。有关更多信息，请参见[如何临时在区域中创建 VNIC \[30\]](#)。

- 使用外部交换机进行 VNIC 之间的通信 – 使用 Oracle Solaris 11.2 反射中继功能，可以将本地 Oracle Solaris 区域或共享相同底层物理 NIC 的 Oracle VM 之间的通信强制为始终发送到物理网络而非主机虚拟交换机。这些实体之间的通信受支持反射中继功能的外部交换机上配置的策略限制。有关更多信息，请参见[“控制基于同一物理端口的 VM 之间的切换” \[71\]](#)。
- 用于监视网络通信统计数据的增强功能 – 可以使用增强的 `dlstat` 和 `flowstat` 命令来有效地监视网络通信统计数据。网络通信统计数据监视增强功能有：
 - 以当前时间显示网络通信统计数据。
 - 根据指定的时间间隔和计数值显示和刷新网络通信统计数据。
 - 根据指定的时间间隔值以每秒的速率显示网络通信统计数据。

有关这些增强功能的更多信息，请参见[“显示网络设备的网络通信统计数据” \[168\]](#)和[“显示流的网络通信统计数据” \[172\]](#)。

- 流配置更改 – 可以使用增强的 `flowadm add-flow` 命令根据其他一些属性及其较新组合在数据链路上配置流，这可帮助您有选择地组织从不同的端口、传输协议和 IP 地址接收的网络包。有关更多信息，请参见[“使用流管理网络资源” \[157\]](#)。

除用于管理流的带宽属性之外，您还可以使用 `flowadm set-flowprop` 命令为流设置 `priority` 属性。可通过设置 `priority` 属性来设置流的优先级。通过新的只读属性 `hwflow`，可以查看流的实例化方式。有关更多信息，请参见[“配置流” \[158\]](#)。

- 显示与 VNIC 关联的多个 MAC 地址 – 可以使用增强的 `dladm show-vmnic` 命令来显示与 VNIC 关联的多个 MAC 地址。有关更多信息，请参见[“显示具有多个 MAC 地址的 VNIC” \[35\]](#)。
- 系统创建的 VNIC – 除您可以使用 `dladm create-vmnic` 命令创建的 VNIC 之外，系统也会创建一些 VNIC，这些 VNIC 称为系统创建的 VNIC。有关更多信息，请参见[“用于配置虚拟网络组件的命令” \[21\]](#)。
- 显示数据链路的物理和虚拟链路状态 – 可以使用 `dladm show-phys` 和 `dladm show-ether` 命令来显示数据链路的物理链路状态。要显示数据链路的虚拟链路状态，可以使用 `dladm show-link` 命令。有关更多信息，请参见[“显示数据链路的物理和虚拟链路状态” \[36\]](#)。
- 显示数据链路属性的生效值 – 增强了 `dladm show-linkprop` 命令以显示数据链路属性的 `EFFECTIVE` 字段。系统会根据资源的可用性、底层设备的功能或与对等方的协商来确定 `EFFECTIVE` 字段的值。生效值不需要与配置的值相同。即使没有为数据链路属性配置值，此属性也可以具有一个生效值。

什么是网络虚拟化和网络资源管理？

网络虚拟化是将硬件网络资源和软件网络资源组合为单一管理单元的过程。此单个管理单元称为虚拟网络。

网络资源管理是指为网络进程管理和分配资源的过程。您可以根据要处理的网络通信量，以不同的方式分配资源。通过根据实际需要管理和分配资源，可以在处理包时提高系统的效率。

与网络资源管理一起有效使用时，网络虚拟化会得到优化。可以向系统和用户提供对硬件和软件网络资源的受控共享，从而提高虚拟网络进程的效率。网络虚拟化及网络资源管理可帮助您管理流控制、提高系统性能，并配置所需的网络利用率以便实现操作系统 (operating system, OS) 虚拟化、实用程序计算和服务器整合。

虚拟网络概述

虚拟网络是一个模拟物理网络的网络，并且是硬件和软件网络资源的组合。虚拟网络是网络虚拟化的最终产品。

虚拟网络可分为两大类：外部和内部。

外部虚拟网络包含作为单个实体由软件进行管理的几个本地网络。传统外部虚拟网络的基本组件是交换机硬件和虚拟局域网 (virtual local area network, VLAN) 软件技术。外部虚拟网络的示例包括大型企业网络和数据中心。有关 VLAN 的更多信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的第 3 章“使用虚拟局域网配置虚拟网络”。

内部虚拟网络由一个系统组成，该系统使用的虚拟机或区域的网络接口至少配置在一个物理网络接口卡 (network interface card, NIC) 上。这些网络接口称为虚拟网络接口卡或虚拟 NIC (Virtual Network Interface Card, VNIC)。这些虚拟机或区域可以相互通信，就像是在同一个本地网络上一样，实际上在一个主机上形成虚拟网络。本文档中的各个章节主要介绍了内部虚拟网络。

一种特殊类型的内部虚拟网络是专用虚拟网络。专用虚拟网络和虚拟专用网络 (virtual private network, VPN) 不同。VPN 会在两个端点系统之间创建一个安全的点对点链路。专用虚拟网络是一个系统中不能通过外部网络访问的虚拟网络。通过在称为 *etherstub* 的伪 NIC 上配置 VNIC 来实现此内部网络与其他外部网络的隔离。有关更多信息，请参见“Etherstub” [14]。

虚拟网络组件

虚拟网络具有以下组件：

- 虚拟网络接口卡 (Virtual Network Interface Card, VNIC)
- 虚拟交换机

- Etherstub
- 区域

虚拟网络接口卡 (Virtual Network Interface Card, VNIC)

VNIC 是行为方式就像配置的物理 NIC 的 L2 实体或虚拟网络设备。可以在底层数据链路上配置 VNIC，以便在多个区域或 VM 之间共享。此外，系统的资源将 VNIC 视为物理 NIC。所有物理以太网接口都支持创建 VNIC。有关如何配置 VNIC 的更多信息，请参见[如何配置 VNIC 和 Etherstub \[22\]](#)。

VNIC 有自动生成的 MAC 地址。根据使用的网络接口，您可以为 VNIC 分配自动生成的 MAC 地址之外的 MAC 地址。有关更多信息，请参见[“修改 VNIC MAC 地址” \[39\]](#)。

虚拟交换机

虚拟交换机是便于虚拟机 (virtual machine, VM) 之间通信的实体。虚拟交换机会循环物理计算机内的虚拟机之间的通信 (VM 间通信)，并且不会在网络上发出此通信。只要在底层数据链路之上创建 VNIC，就会隐式创建虚拟交换机。配置了 VM 的 VNIC 需要位于相同的 VLAN 或 VXLAN 上才能进行 VM 间通信。可通过 EVS 管理虚拟交换机。有关 EVS 的信息，请参见[第 5 章 关于弹性虚拟交换机](#)。

按照以太网设计，如果一个交换机端口接收从连接到该端口的主机传出的包，该包无法到达同一端口上的目标。对于使用虚拟网络配置的系统来说，这种以太网设计是一种限制，因为虚拟网络共享同一个 NIC。通过使用虚拟交换机会克服这种以太网设计限制，从而使 VM 能够相互通信。

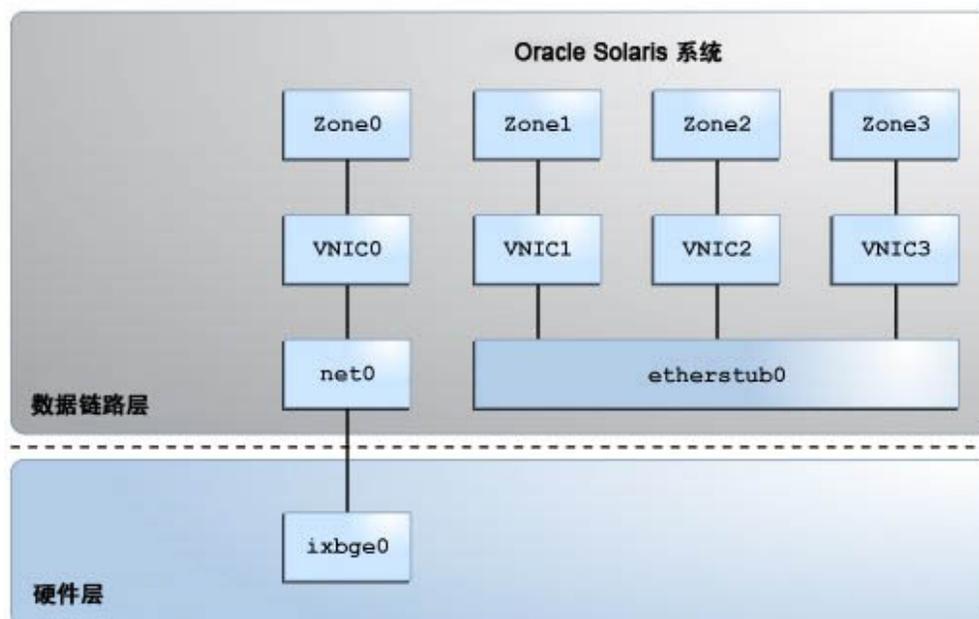
在某些情况下，系统中 VM 之间的通信可能要求使用交换机。例如，VM 之间的通信可能需要遵守交换机上配置的访问控制列表 (access control list, ACL)。缺省情况下，交换机不能使用接收包的同一端口发送包。因此，要在使用外部交换机的 VM 之间进行通信，需在交换机上启用反射中继。反射中继使交换机能够使用接收包的同一端口转发包。有关更多信息，请参见[反射中继](#)。

Etherstub

etherstub 是在 Oracle Solaris 网络栈的数据链路层 (L2) 上配置的伪以太网 NIC。您可以在 *etherstub* 上（而不是在物理 NIC 上）创建 VNIC。使用 *etherstub*，您可以构建同时与系统上的其他虚拟网络和外部网络隔离的专用虚拟网络。例如，可以使用 *etherstub* 创建一个仅供开发者（而不是整个网络）访问的网络环境。

下图显示了一个基于 *etherstub* 的专用虚拟网络。

图 1-1 专用虚拟网络



此图显示了在其上配置 VNIC1、VNIC2 和 VNIC3 的 etherstub0。每个 VNIC 都分配给一个区域。不能通过外部网络访问基于 etherstub 的专用虚拟网络。有关更多信息，请参见[如何配置专用虚拟网络 \[31\]](#)。

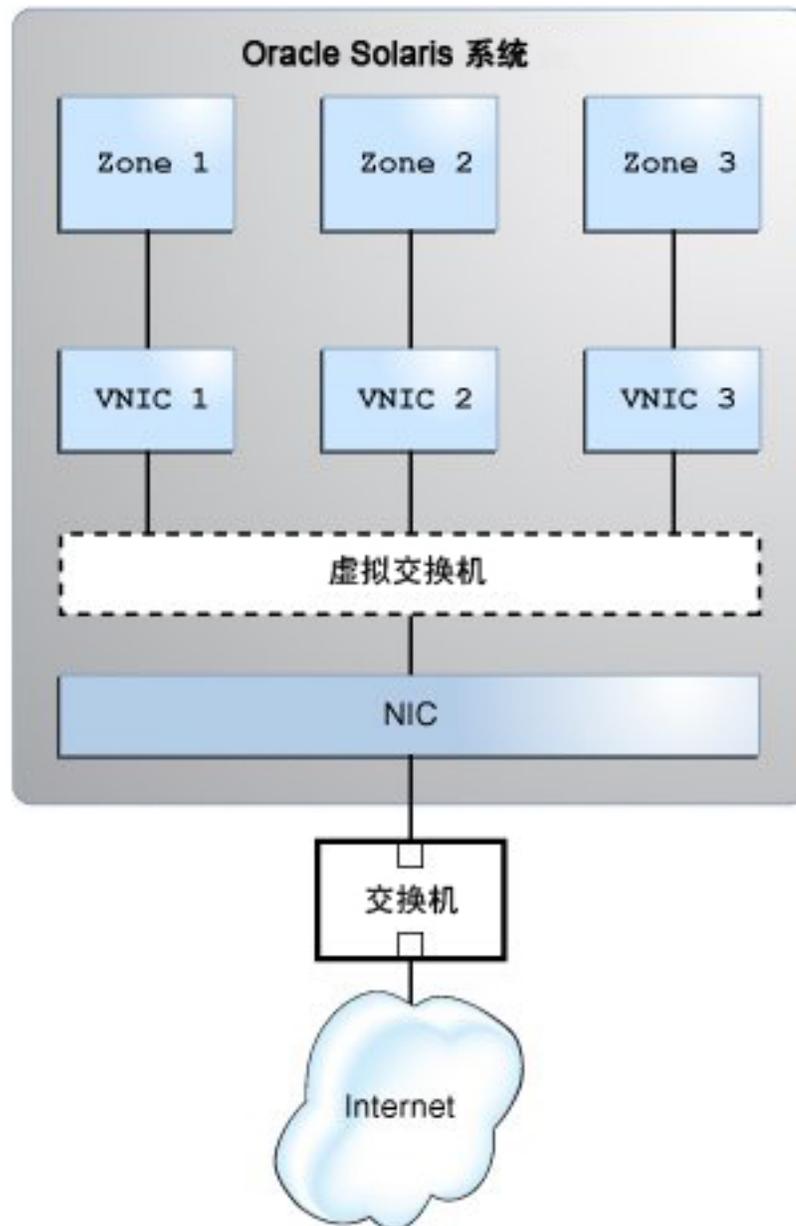
区域

区域是在 Oracle Solaris 操作系统的一个单独实例中创建的一个虚拟化的操作系统环境。区域提供了用于运行应用程序的受保护的隔离环境。Etherstub 和 VNIC 仅是 Oracle Solaris 的虚拟化功能的一部分。通过分配 VNIC 或 etherstub 供 Oracle Solaris 区域使用，您可以在单个系统内创建一个网络。有关区域的更多信息，请参见《[Oracle Solaris Zones 介绍](#)》。

虚拟网络的工作方式

下图显示了虚拟网络及其组件在系统中的工作方式。

图 1-2 虚拟网络的工作方式



此图显示具有一个 NIC 的单个系统。该 NIC 配置了三个 VNIC。每个 VNIC 都分配给一个区域。Zone 1、Zone 2 和 Zone 3 是三个配置为在系统中使用的区域。区域使用各自的 VNIC 相互通信以及与外部网络通信。这三个 VNIC 通过虚拟交换机连接到底层物理 NIC。虚拟交换机的功能等效于物理交换机的功能，因为二者都提供与系统的连接。

配置虚拟网络后，区域将通信发送到外部主机的方式与没有虚拟网络的系统相同。通信从区域经由 VNIC 到达虚拟交换机，然后到达物理接口，由物理接口将数据发送到网络。

如果配置为区域的所有 VNIC 都属于相同的 VLAN，则还能在系统中的区域之间交换通信。例如，包从 Zone 1 传递到其专用 VNIC 1。然后，通信经过虚拟交换机到达 VNIC 3。接着，VNIC 3 将通信传递到 Zone 3。该通信从未离开过系统，因此永远不会违反以太网限制。

您也可以创建一个基于 etherstub 的虚拟网络。Etherstub 完全基于软件，不需要有网络接口作为虚拟网络的基础。

Oracle 还提供了 Oracle Enterprise Manager Ops Center 来管理网络虚拟化的某些方面，例如在虚拟数据中心内创建虚拟网络的能力。有关 Oracle Enterprise Manager Ops Center 的更多信息，请参见位于 <http://www.oracle.com/pls/topic/lookup?ctx=oc122&id=OPCCM> 的文档库。

使用虚拟可扩展局域网

虚拟可扩展局域网 (Virtual Extensible Local Area Network, VXLAN) 是一项网络虚拟化技术，此技术为虚拟网络提供了可扩展性和网络隔离。VXLAN 解决了虚拟局域网 (virtual local area network, VLAN) 的 4K 限制，并且还减少了物理基础结构（例如交换机）上的虚拟化需求。有关更多信息，请参见第 3 章 [使用虚拟可扩展局域网配置虚拟网络](#)。

使用边缘虚拟桥接

通过边缘虚拟桥接 (edge virtual bridging, EVB)，主机可以与外部交换机交换与系统上的虚拟链路相关的信息。EVB 用于交换有关端口后的所有虚拟链路的信息，而 DCB 用于交换有关端口的信息。有关 EVB 的更多信息，请参见第 4 章 [使用边缘虚拟桥接管理服务-网络边缘虚拟化](#)。

谁应实现虚拟网络？

如果您需要整合 Oracle 的 Sun 服务器上的资源，可以考虑构建虚拟网络。通过整合少数几个服务器上的各种应用程序，可以提高可用资源的利用率。然后，可以使用虚拟网络在这些应用程序之间提供连接。

ISP 的集运商、电信公司和大型金融机构可以根据下面的硬件资源整合其服务器：

- 支持高带宽和各种硬件（例如，对 NIC 环和虚拟功能 (virtual function, VF) 的支持) 的功能强大的 NIC
- 具有更先进的随机访问存储器 (random access memory, RAM) 和中央处理器 (central processing unit, CPU) 的功能强大的物理计算机

您可以用包含多个区域或虚拟机的单个系统取代多个系统，而不会明显损失隔离性、安全性和灵活性。

有关网络虚拟化优点的演示，请参见 [Consolidating the Data Center With Network Virtualization \(http://download.oracle.com/otndocs/tech/OTN_Demos/data-center-consolidation.html\)](http://download.oracle.com/otndocs/tech/OTN_Demos/data-center-consolidation.html)（使用网络虚拟化合并数据中心）。

网络资源管理概述

在 Oracle Solaris 中，可以通过管理网络资源来更轻松且动态地实现服务质量 (quality of service, QoS)。网络资源管理相当于为通信流量创建专用通道。当您组合不同的资源以提供特定类型的网络包时，这些资源就形成一个用于这些包的网络通道。可以为每个网络通道以不同的方式指定资源。例如，您可以为网络通信流量最大的通道分配更多资源。通过配置根据实际需要分配资源的网络通道，可以提高系统处理网络包的效率。有关网络通道的更多信息，请参见“[监视数据链路和流的网络通信统计数据概述](#)” [165]。

使用下面的网络资源可以提高系统处理包的效率：

- 带宽 – 您可以根据数据链路支持的网络进程的实际需要来限制数据链路的带宽。
- 优先级 – 您可以设置处理包的优先顺序。优先级较高的包的延迟将减少，因为这些包在其他包之前得到处理。
- NIC 环 – 如果 NIC 支持环分配，可将其传送和接收环指定为专供数据链路使用。有关更多信息，请参见“[管理 NIC 环](#)” [144]。
- CPU 池 – CPU 池是使用特定区域创建的并与之相关联。可以将这些池进一步分配给数据链路以便管理其相关联的区域的网络进程。有关更多信息，请参见“[管理池和 CPU](#)” [152]。
- CPU – 在具有多个 CPU 的系统上，您可以分配给定数量的 CPU 专用于特定网络进程。有关更多信息，请参见“[管理池和 CPU](#)” [152]。

可通过使用数据链路属性或流来管理系统上的网络资源。

通过使用数据链路属性进行网络资源管理

通过使用数据链路属性管理网络资源可以提高系统处理包的效率。您可以在创建链路时设置数据链路属性。也可以稍后设置数据链路属性，例如：在了解资源使用情况一段时间并确定如何更好地分配资源之后再设置这些属性。通过设置与网络资源相关的数据链路属性，您可以确定给定资源在多大程度上可用于网络进程。资源分配过程既适用于虚拟网络，也适用于物理网络。有关数据链路属性及其配置方式的更多信息，请参见[“使用数据链路属性管理网络资源” \[143\]](#)。

通过使用流管理网络资源

流是根据单个属性或属性组合对网络包进行分类的定制方式。作为创建流的基础的属性是从网络包头信息派生的。针对网络资源管理设置数据链路属性后，可使用流来进一步控制如何使用资源来处理网络包。也可以单独使用流来管理网络资源，而无需设置数据链路属性。

使用流管理资源包含以下步骤：

1. 根据单个属性或属性组合创建流。
2. 通过设置与网络资源相关的属性，定制流的资源使用。当前，只有带宽和优先级属性才能设置为流。

有关配置流的更多信息，请参见[“使用流管理网络资源” \[157\]](#)。

网络资源管理的益处

通过使用网络资源管理，您可以对单个系统上的数据通信进行隔离、设置优先级、跟踪和控制，而无需使用复杂的 QoS 规则定义。

网络资源管理有助于完成以下任务：

- 置备网络
- 建立服务级别协议
- 客户计费
- 诊断安全问题

创建和管理虚拟网络

本章介绍了在单个系统中配置虚拟网络组件、构建虚拟网络以及管理 VNIC 的任务。有关虚拟网络的介绍，请参见第 1 章 [网络虚拟化和网络资源管理简介](#)。

本章包含以下主题：

- [“配置虚拟网络的组件” \[21\]](#)
- [“构建虚拟网络” \[25\]](#)
- [“管理 VNIC” \[34\]](#)
- [“将单根 I/O 虚拟化与 VNIC 一起使用” \[45\]](#)

配置虚拟网络的组件

在 Oracle Solaris 中，VNIC 和 etherstub 是虚拟网络的基本组件。本节介绍了用于配置这些组件以便构建虚拟网络的步骤。有关这些组件的说明，请参见[“虚拟网络组件” \[13\]](#)。

用于配置虚拟网络组件的命令

要创建 VNIC，请使用 `dladm create-vnic` 命令。

```
# dladm create-vnic -l link [-v vid] VNIC
```

<i>link</i>	配置 VNIC 时所基于的链路的名称。
<i>vid</i>	VNIC 的 VLAN ID（如果要将 VNIC 创建为 VLAN）。要配置具有 VLAN ID 的 VNIC，请参见 如何配置带有 VLAN ID 的 VNIC [24] 。有关 VLAN 的更多信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的第 3 章“使用虚拟局域网配置虚拟网络”。
<i>VNIC</i>	VNIC 的名称。有关如何创建定制名称的准则，请参见《在 Oracle Solaris 11.2 中配置和管理网络组件》中的“有效链路名称的规则”。

您可以为 VNIC 配置其他属性，例如要与 VNIC 关联的 MAC 地址和 CPU。有关这些属性的列表，请参见 [dladm\(1M\)](#) 手册页。某些属性修改只适用于 VNIC。例如，通过 `dladm create-vnic` 命令，您可以配置一个 MAC 地址并指定 VLAN ID 以将 VNIC 创建为 VLAN。不过，您不能使用 `dladm create-vlan` 命令直接为 VLAN 配置 MAC 地址。

一次只能在一个数据链路上创建一个 VNIC。与数据链路一样，VNIC 具有您可以按需进一步配置的链路属性。有关不同类型的链路属性的信息，请参见“[通过使用数据链路属性进行网络资源管理](#)” [19]。

除了您可以通过使用 `dladm create-vnic` 命令创建的 VNIC 之外，系统也会创建 VNIC（称为系统创建的 VNIC），用以帮助实现 Oracle VM Server for SPARC vnet 的虚拟网络 I/O。系统创建的 VNIC 遵循命名约定 `<entity>-<name>`，其中 `entity` 是指创建了 VNIC 的系统实体，`name` 是指在系统实体内的 VNIC 名称。用户创建的 VNIC 名称不能包含连字符 (-)。只有系统创建的 VNIC 会包含连字符 (-)，这可帮助您区分系统创建的 VNIC 和用户创建的 VNIC。无法修改、重命名、激活或删除系统创建的 VNIC。有关更多信息，请参见《[Oracle VM Server for SPARC 3.1 管理指南](#)》。

可以使用 `dlstat` 和 `snoop` 命令监视系统创建的 VNIC 上的网络通信。您还可以使用 `flowadm` 命令基于系统创建的 VNIC 来创建流。流不仅可帮助您管理网络资源，还可帮助您监视网络通信统计数据。您可以使用 `flowstat` 命令监视流的网络通信统计数据。有关流的更多信息，请参见“[配置流](#)” [158]。

要创建 etherstub，请使用 `dladm create-etherstub` 命令。

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

其中 `etherstub` 是指要创建的 etherstub 的名称。

▼ 如何配置 VNIC 和 Etherstub

VNIC 将虚拟网络连接到外部网络。VNIC 还能够使区域通过 VNIC 自动创建的虚拟交换机相互通信。对于托管着区域间的内部通信以及区域与外部 LAN 和 Internet 之间的通信的虚拟网络，每个区域都必须具有自己的 VNIC。因此，有多少属于虚拟网络的区域，就必须重复多少次该过程。

1. 成为管理员。
有关更多信息，请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“[使用所指定的管理权限](#)”。
2. （可选）创建 etherstub。

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

只有创建专用虚拟网络时才需要执行此步骤。有关专用虚拟网络的说明，请参见“[虚拟网络概述](#)” [13]。有关如何配置专用虚拟网络的更多信息，请参见[如何配置专用虚拟网络](#) [31]。

与数据链路一样，您可以采用任何对您的网络设置有意义的方式来命名 etherstub。有关如何创建定制名称的准则，请参见《在 Oracle Solaris 11.2 中配置和管理网络组件》中的“有效链路名称的规则”。

3. 创建 VNIC。

```
# dladm create-vnic -l link [-v vid] VNIC
```

如果要为专用虚拟网络创建 VNIC，请使用 *etherstub* 替代 *link*。只有要将 VNIC 创建为 VLAN 时，才在命令中包括 -v 选项。有关将 VNIC 创建为 VLAN 的更多信息，请参见[如何配置带有 VLAN ID 的 VNIC \[24\]](#)。

4. 在 VNIC 上创建 IP 接口。

```
# ipadm create-ip interface
```

interface 您在上一步中创建的 VNIC。

5. 为 VNIC 接口指定一个静态 IP 地址。

```
# ipadm create-addr -a address interface
```

-a *address* 指定 IP 地址，该地址可以采用无类域间路由 (Classless Inter-Domain Routing, CIDR) 表示法。

静态 IP 地址可以为 IPv4 或 IPv6 地址。有关更多信息，请参见《在 Oracle Solaris 11.2 中配置和管理网络组件》中的“如何配置 IPv4 接口”。

6. (可选) 验证已创建的 VNIC。

```
# dladm show-link
```

例 2-1 配置 VNIC

此示例说明了如何在数据链路 net0 上配置 vnic1。

```
# dladm create-vnic -l net0 vnic1
# ipadm create-ip vnic1
# ipadm create-addr -a 192.168.0.10/24 vnic1
# dladm show-link
LINK          CLASS      MTU   STATE   OVER
net0          phys      1500  up      --
vnic1         vnic      1500  up      net0
```

▼ 如何配置带有 VLAN ID 的 VNIC

您可以配置带 VLAN ID 的 VNIC 来托管 VLAN 通信。如果 VNIC 需要属于某个 VLAN 并接收该 VLAN 的通信，则您需要将该 VLAN 的 VLAN ID 指定给 VNIC。您还可以设置链路属性 `vlan-announce` 以将每个 VNIC 的 VLAN 配置传播到网络。

与常规 VLAN 链路不同，配置为 VLAN 的 VNIC 有自己的 MAC 地址。有关常规 VLAN 的信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的第 3 章“使用虚拟局域网配置虚拟网络”。

此过程仅包含用于创建带有 VLAN ID 的 VNIC 的步骤以及用于设置可使 VNIC 提供 VLAN 通信服务的相应属性的步骤。尽管在启用 `vlan-announce` 属性时会自动更新中间端口和交换机，但是，必须分别配置中间端口和交换机以便在这些点上定义 VLAN。

1. 成为管理员。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 创建带有 VLAN ID 的 VNIC。

```
# dladm create-vnic -l link -v vid VNIC
```

3. (可选) 向网络广播 VNIC 的 VLAN 配置。

```
# dladm set-linkprop -p vlan-announce=gvrp link
```

通过此步骤可启用 GARP VLAN 注册协议 (GARP VLAN Registration Protocol, GVRP) 客户机系统，该系统自动在连接的交换机中注册 VLAN ID。缺省情况下，`vlan-announce` 属性设置为 `off`，表示不会将任何 VLAN 广播消息发送到网络。将该属性设置为 `gvrp` 后，会传播该链路的 VLAN 配置以启用网络设备的自动 VLAN 端口配置。然后，这些设备可以接受 VLAN 通信并将其转发。有关 GVRP 的更多信息，请参见《Sun Ethernet Fabric Operating System》（《Sun 以太网光纤操作系统》）中的“Configuring GVRP”（“配置 GVRP”）<http://docs.oracle.com/cd/E19285-01/html/E20769/z40015861393155.html#scrolltoc>。

4. (可选) 设置 `gvrp-timeout` 属性以配置各次 VLAN 广播之间的等待时间。

```
# dladm set-linkprop -p gvrp-timeout=time link
```

`time` 指 `gvrp-timeout` 属性的值（毫秒）。缺省值为 250 毫秒。负载很重的系统在重新广播 VLAN 信息时需要一个较短的时间间隔。通过该属性可调整时间间隔。

5. (可选) 显示属性 `vlan-announce` 和 `gvrp-timeout` 的值。

```
# dladm show-linkprop -p vlan-announce,gvrp-timeout
```

例 2-2 将 VNIC 配置为 VLAN

此示例说明了如何在数据链路 net0 上创建一个带有 VLAN ID 123 且名为 vnic0 的 VNIC，以及如何向网络公布 VLAN 配置。

```
# dladm create-vnic -l net0 -v 123 vnic0
# dladm set-linkprop -p vlan-announce=gvrp net0
# dladm set-linkprop -p gvrp-timeout=250 net0
# dladm show-linkprop -p vlan-announce,gvrp-timeout net0
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net0	vlan-announce	rw	gvrp	gvrp	off	off,gvrp
net0	gvrp-timeout	rw	250	250	250	100-100000

该输出显示了以下信息：

LINK	物理数据链路，由名称予以标识。
PROPERTY	链路的属性。一个链路可以有几个属性。
PERM	属性的权限，可以是以下权限之一： <ul style="list-style-type: none"> ▪ ro 是指链路属性的只读权限。 ▪ rw 是指链路属性的读取和写入权限。
VALUE	当前（或持久性的）链路属性值。如果未设置此值，此值将显示为 --。在未知情况下，此值将显示为 ?。
DEFAULT	链路属性的缺省值。如果链路属性没有缺省值，则显示 --。
POSSIBLE	链路属性可具有的值的逗号分隔列表。如果可能的值未知或无限制，则显示 --。

构建虚拟网络

必须创建区域才能构建虚拟网络。您可以根据系统支持能力创建所需的任意数量的区域。每个区域都有自己的虚拟接口。系统中的区域可相互通信。虚拟网络作为一个整体连接到大型外部网络上的目标。

要构建虚拟网络，您必须配置 etherstub 或 VNIC，并配置区域。尽管这些是独立的过程集，但是必须同时执行才能完成虚拟网络的构建。

本节中的过程基于以下假设：

- 系统上的虚拟网络包括三个区域并且这三个区域分别处于不同的配置阶段。第一个区域是作为新区域创建的，第二个区域已存在于系统上并需要重新配置才能使用 VNIC，第三个区域被指定为专用虚拟网络并需要启用才能发送跨系统的网络通信。

- 系统物理接口的 IP 地址配置为 192.168.3.70。
- 路由器的 IP 地址为 192.168.3.25。

构建虚拟网络时，一些步骤需在全局区域中执行，一些步骤需在非全局区域中执行。为清楚起见，每个步骤后面的示例中的提示指明了在哪个区域中发出特定命令。但是，提示显示的实际路径可能有所不同，具体取决于特定于您系统的提示。

有关配置虚拟网络的演示，请参见 [Configuring a Virtual Network in Oracle Solaris - Part 1](http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part1/VirtualDemo_Part1.htm) (http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part1/VirtualDemo_Part1.htm) (在 Oracle Solaris 中配置虚拟网络 - 第 1 部分) 和 [Configuring a Virtual Network in Oracle Solaris - Part 2](http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part2/VirtualDemo_Part2.htm) (http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part2/VirtualDemo_Part2.htm) (在 Oracle Solaris 中配置虚拟网络 - 第 2 部分)。

▼ 如何为虚拟网络配置区域

此过程介绍了如何使用新 VNIC 配置新区域。请注意，此过程仅包含与网络虚拟化相关的步骤。有关如何配置区域的更多信息，请参见《[创建和使用 Oracle Solaris 区域](#)》中的第 1 章“[如何规划和配置非全局区域](#)”。

1. 成为管理员。
有关更多信息，请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“[使用所指定的管理权限](#)”。
2. 配置 VNIC。
有关更多信息，请参见[如何配置 VNIC 和 Etherstub \[22\]](#)。

3. 创建区域。

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

zone 指区域的名称。

确保指定先前作为区域的物理接口创建的 VNIC。缺省情况下，区域的 ip-type 参数设置为 exclusive。

4. 验证并提交您已实现的更改，然后退出该区域。

```
zonecfg:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

5. 安装区域。

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

6. 启动区域。

```
global# zoneadm -z ZONE boot
```

7. 在区域完全启动后，登录到该区域。

```
global# zlogin -C zone
```

8. 根据提示指定信息。

您可以通过从选项列表中选择来指定大部分信息。通常，缺省选项足以满足需求。要配置虚拟网络，必须指定或验证以下信息：

- 区域的主机名，例如 zone1
- 区域的 IP 地址（它基于区域的 VNIC 的 IP 地址）
- 是否应启用 IPv6
- 具有虚拟网络的系统是否为子网的一部分
- IP 地址的网络掩码
- 缺省路由，可以是在其上构建虚拟网络的物理接口的 IP 地址

提供所需的信息后，该区域将重新启动。

另外，您也可以配置一个专用 IP 区域，该区域带有称为 anet 资源的自动 VNIC。有关更多信息，请参见《[创建和使用 Oracle Solaris 区域](#)》中的“[如何配置区域](#)”。

例 2-3 为虚拟网络配置区域

在此示例中，为虚拟网络创建了 zone1，并将 vnic1 连接为物理接口。请注意，只列出了与创建虚拟网络相关的区域参数。

```
global # zonecfg -z zone1
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/export/home/zone1
zonecfg:zone1> set autoboot=true
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
```

```
global# zoneadm -z zone1 install
.
.
.
global# zoneadm -z zone1 boot
```

```
global# zlogin -C zone1
```

为配置网络，提供了以下信息：

```
Hostname: zone1
IP address: 192.168.3.80
System part of a subnet: Yes
Netmask: 255.255.255.0
Enable IPv6: No
Default route: 192.168.3.70
Router IP address: 192.168.3.25
```

▼ 如何重新配置区域以使用 VNIC

此过程中所提及的区域指的是虚拟网络中的第二个区域。该区域已存在，但其当前配置阻止它成为虚拟网络的一部分。具体地说，该区域的 IP 类型是共享类型，而其当前接口是 net0。这两个配置必须更改。

1. 成为管理员。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 创建 VNIC。

```
global# dladm create-vnic -l link VNIC
```

您将在此过程的后半部分配置 VNIC 的接口。

3. 将区域的 IP 类型从 shared 更改为 exclusive。

```
global# zonecfg -z zone
zonecfg:zone> set ip-type=exclusive
```

4. 更改区域的接口以使用 VNIC。

```
zonecfg:zone> remove net physical=NIC
zonecfg:zone> add net
zonecfg:zone:net> set physical=VNIC
zonecfg:zone:net> end
```

5. 验证并提交您已实现的更改，然后退出该区域。

```
zonecfg:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

6. 重新引导区域。

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

7. 登录到区域。

```
global# zlogin zone
```

8. 在区域中，基于现已指定给区域的 VNIC 创建 IP 接口。

```
zone# ipadm create-ip interface
```

9. 为 VNIC 配置静态 IP 地址或动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP) IP 地址。

- 指定静态 IP 地址。

```
zone# ipadm create-addr -a address interface
```

`-a address` 指定 IP 地址，该地址可以采用 CIDR 表示法。

- 指定 DHCP IP 地址。

```
zone# ipadm create-addr -T dhcp interface
```

10. 退出区域。

```
zone# exit
```

11. 在全局区域中，将地址信息添加到 `/etc/hosts` 文件中。

例 2-4 重新配置区域以使用 VNIC

在此示例中，`zone2` 已作为共享区域存在。此外，该区域使用的是系统的主接口，而不是虚拟链路。您需要修改 `zone2` 才能使用 `vnic2`。要使用 `vnic2`，必须首先将 `zone2` 的 IP 类型更改为 `exclusive`。请注意，某些输出已截断，以重点显示与虚拟网络相关的信息。

```
global# dladm create-vnic -l net0 vnic2

global# zonecfg -z zone2
zonecfg:zone2> set ip-type=exclusive
zonecfg:zone2> remove net physical=net0
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vnic2
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z zone2 reboot

global# zlogin zone2
zone2# ipadm create-ip vnic2
zone2# ipadm create-addr -a 192.168.3.85/24 vnic2
ipadm: vnic2/v4
```

```
zone2# exit

global# pfedit /etc/hosts
#
::1          localhost
127.0.0.1   localhost
192.168.3.70 loghost    #For net0
192.168.3.80 zone1     #using vnic1
192.168.3.85 zone2     #using vnic2
```

▼ 如何临时在区域中创建 VNIC

通过将链路指定为 *zonelink*，可以从全局区域直接在非全局区域中创建 VNIC。此方法直接在非全局区域的名称空间中创建 VNIC。-t 选项用于指定 VNIC 是临时的。临时 VNIC 将持续到下一次重新引导区域之前。全局区域和其他非全局区域还可以具有同名的 VNIC。使用此方法只能临时创建 VNIC。

除临时创建 VNIC 之外，您还可以临时创建 VLAN 和基于 InfiniBand 的 IP (IP over InfiniBand, IPoIB) 分区。有关完整说明，请参见 [dladm\(1M\)](#) 手册页。

1. 成为管理员。
有关更多信息，请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“使用所指定的管理权限”。

2. 从全局区域创建并引导非全局区域。

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

3. 为非全局区域创建临时的 VNIC。

```
global# dladm create-vnic -t -l link zone/VNIC
```

-t 指定 VNIC 为临时 VNIC。临时 VNIC 将持续到下一次重新引导区域之前。如果在非全局区域的名称空间中创建 VLAN，必须指定此选项。

-l 指定链路，可以是物理链路或 etherstub。

有关将用来从全局区域在非全局区域中创建 VLAN 或 IPoIB 分区的命令语法的示例，请参见例 2-5 “[在区域中临时创建 VNIC、VLAN 和基于 IB 的 IP 分区](#)”。

4. 确认已在区域中创建了 VNIC。

```
global# dladm show-link -Z
```

5. 登录到区域。

```
global# zlogin zone
```

6. 确认已成功创建了 VNIC。

```
zone# dladm show-link
```

例 2-5 在区域中临时创建 VNIC、VLAN 和基于 IB 的 IP 分区

以下示例说明了如何从全局区域在非全局区域中创建名为 vnic1 的 VNIC。

```
global# zoneadm -z zone1 boot
global# dladm create-vnic -t -l net0 zone1/vnic1
global# dladm show-link -Z
LINK                ZONE    CLASS    MTU    STATE    OVER
net0                 global  phys     1500   up       --
zone1/vnic1         zone1   vnic     1500   down    net0
```

以下示例显示了 zone1 的 dladm show-link 命令输出。

```
zone1# dladm show-link
LINK                CLASS    MTU    STATE    OVER
vnic1               vnic     1500   down    ?
```

以下示例说明了如何从全局区域在非全局区域中创建名为 vlan3 的 VLAN。

```
global# dladm create-vlan -t -l net0 -v 3 zone1/vlan3
```

-v 选项指定基于以太网链路的 VLAN 的 VLAN-ID。

以下示例说明了如何从全局区域在非全局区域中创建名为 part1 的 IPoIB 分区。

```
global# dladm create-part -t -l net1 -P FFFF zone1/part1
```

-P 选项指定用于创建分区链路的分区密钥。

▼ 如何配置专用虚拟网络

此过程说明了如何创建专用虚拟网络并使其能够跨系统发送网络通信。尽管该区域是虚拟网络的一部分，但外部系统无法对其进行访问。要使隔离区域能够跨系统发送网络通信，则必须使用网络地址转换 (network address translation, NAT)。NAT 将 VNIC 的专用 IP 地址转换为物理网络接口的可路由 IP 地址。但是，专用 IP 地址对外部网络不可见。有关 NAT 的更多信息，请参见《在 Oracle Solaris 11.2 中确保网络安全》中的“使用 IP 过滤器的 NAT 功能”。

是否使用 etherstub 是虚拟网络与专用虚拟网络之间的主要区别。在专用虚拟网络中，指定给区域的 VNIC 是在 etherstub 上配置的，并且与流经系统的网络通信隔离开来。

以下过程假设该区域已存在，但当前不包含任何关联接口。

1. 成为管理员。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 创建 etherstub。

```
global# dladm create-etherstub etherstub
```

3. 在 etherstub 上创建一个 VNIC。

```
global# dladm create-vnic -l etherstub VNIC
```

您将在此过程的后半部分配置 VNIC 的接口。

4. 将 VNIC 指定给该区域。

```
global# zonecfg -z ZONE
zonecfg:ZONE> add net
zonecfg:ZONE:net> set physical=VNIC
zonecfg:ZONE:net> end
```

5. 验证并提交您已实现的更改，然后退出该区域。

```
zonecfg:ZONE> verify
zonecfg:ZONE> commit
zonecfg:ZONE> exit
```

6. 重新引导区域。

```
global# zoneadm -z ZONE reboot
```

7. 登录到区域。

```
global# zlogin ZONE
```

8. 在该区域中，通过现已指定给该区域的 VNIC 创建 IP 接口。

```
zone# ipadm create-ip interface
```

9. 为 VNIC 配置静态 IP 地址或 DHCP IP 地址。

- 指定静态 IP 地址。

```
zone# ipadm create-addr -a address interface
```

- 指定 DHCP IP 地址。

```
zone# ipadm create-addr -T dhcp interface
```

- 退出区域。

```
zone# exit
```

- 在全局区域中，将地址信息添加到 `/etc/hosts` 文件中。

- 在全局区域中，设置主接口以执行 IP 转发。

```
global# ipadm set-ifprop -p forwarding=on -m ipv4 primary-interface
```

注 - 在 Oracle Solaris 中，主接口是 NIC 的物理数据链路。

- 在全局区域中，在 `/etc/ipnat.conf` 文件中为主接口配置网络地址转换 (network address translation, NAT)。

- 启动 IP 过滤服务以启用 NAT。

```
global# svcadm enable network/ipfilter
```

- 重新引导区域。

```
global# zoneadm -z ZONE reboot
```

例 2-6 配置专用虚拟网络

在本示例中，将 `zone3` 配置为要隔离的专用网络。同时，将 NAT 和 IP 转发配置为允许专用虚拟网络向主机外部发送包，但仍对外部网络隐藏其专用地址。已使用专用 IP 类型配置了该区域。但是，不为该区域指定 IP 接口。

```
global# dladm create-etherstub ether0
global# dladm create-vnic -l ether0 vnic3
global# zonecfg -z zone3
zonecfg:zone3> add net
zonecfg:zone3:net> set physical=vnic3
zonecfg:zone3:net> end
zonecfg:zone3> verify
zonecfg:zone3> commit
zonecfg:zone3> exit

global# zoneadm -z zone3 reboot
global# zlogin zone3
zone3# ipadm create-ip vnic3
zone3# ipadm create-addr -a 192.168.0.10/24 vnic3
ipadm: vnic3/v4
zone3# exit

global# pfedit /etc/hosts
::1          localhost
127.0.0.1   localhost
```

```

192.168.3.70    loghost    #For net0
192.168.3.80    zone1     #using vnic1
192.168.3.85    zone2     #using vnic2
192.168.0.10    zone3     #using vnic3

global# ipadm set-ifprop -p forwarding=on -m ipv4 vnic3

global# pfedit /etc/ipf/ipnat.conf
map vnic3 192.168.0.0/24 -> 0/32 portmap tcp/udp auto
map vnic3 192.168.0.0/24 -> 0/32

global# svcadm enable network/ipfilter
global# zoneadm -z zone3 reboot

```

管理 VNIC

本节介绍了在执行完基本配置之后可以对 VNIC 执行的任务。有关如何执行 VNIC 的基本配置的信息，请参见[如何配置 VNIC 和 Etherstub \[22\]](#)。

您可以修改 VNIC 的 VLAN ID、MAC 地址和底层数据链路。修改底层数据链路意味着将 VNIC 移动到另一个数据链路。您可以全局修改数据链路上的所有 VNIC 的属性，也可以有选择地仅修改指定 VNIC 的属性。

本节包含以下主题：

- [“显示 VNIC” \[34\]](#)
- [“修改 VNIC 的 VLAN ID” \[38\]](#)
- [“修改 VNIC MAC 地址” \[39\]](#)
- [“迁移 VNIC” \[41\]](#)
- [“删除 VNIC” \[42\]](#)

显示 VNIC

要获取有关您系统上的 VNIC 的信息，请使用 `dladm show-vnic` 命令。

例 2-7 显示系统上的 VNIC

```

# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic1     net0      1000      2:8:20:c2:39:38  random           123
vnic2     net0      1000      2:8:20:5f:84:ff  random           456

```

该输出显示了以下信息：

LINK	虚拟数据链路，由名称予以标识。
OVER	配置 VNIC 时所基于的物理或虚拟数据链路。
SPEED	VNIC 的最大速度（兆位/秒）。
MACADDRESS	VNIC 的 MAC 地址。
MACADDRTYPE	VNIC 的 MAC 地址类型，可以是以下类型之一： <ul style="list-style-type: none"> ▪ random – 为 VNIC 指定的随机地址 ▪ factory – VNIC 使用的 NIC 的出厂 MAC 地址 ▪ fixed – 用户指定的 MAC 地址
VID	VNIC 的 VLAN ID。

任何用于显示数据链路的相关信息的 `dladm` 命令都可以用来显示系统上的 VNIC（如果存在）的信息。例如，`dladm show-link` 命令用于显示 VNIC 以及其他数据链路。您可以使用 `dladm show-linkprop` 命令来显示 VNIC 的属性。

要获取有关单个 VNIC 的数据链路属性的信息，请使用下面的命令语法指定 VNIC：

```
# dladm show-linkprop [-p property] vnic
```

例 2-8 显示连接到区域的 VNIC

在此示例中，显示了连接到区域的主数据链路和 VNIC 的信息。主数据链路 `net0` 连接到全局区域。VNIC `vnic1` 和 `vnic2` 分别连接到 `zone1` 和 `zone2`。

```
# dladm show-link -Z
LINK          ZONE    CLASS  MTU   STATE  OVER
net0          global  phys   1500  up     --
zone1/vnic1   zone1   vnic   1500  up     net0
zone2/vnic2   zone2   vnic   1500  up     net0
```

显示具有多个 MAC 地址的 VNIC

对于 Oracle VM Server for SPARC 中由系统创建的 VNIC 和 Oracle Solaris 内核区域中的 `anet` 资源，有多个 MAC 地址与之关联。在 Oracle VM Server for SPARC 中，您需要创建一个具有 `alt-mac-addr` 属性的 `vnet` 以支持来宾域内的 VNIC 和区域。在这种情况下，系统将自动创建一个具有多个 MAC 地址的 VNIC。这些 MAC 地址是从您创建的 `vnet` 获取的。有关更多信息，请参见 [《Oracle VM Server for SPARC 3.1 管理指南》](#)。

要支持内核区域内的区域或 VNIC，您可以配置具有多个 MAC 地址的 `anet` 资源。可以使用 `zonecfg` 命令向在内核区域中为网络访问创建的 `anet` 资源指定多个 MAC 地址。有

关更多信息，请参见 [solaris-kz\(5\)](#) 手册页。有关配置内核区域的信息，请参见《[创建和使用 Oracle Solaris 内核区域](#)》。

当有多个 MAC 地址与 VNIC 关联时，虚拟网络驱动程序使用其中一个 MAC 地址。您可以使用其余的 MAC 地址在内核区域或来宾域内创建 VNIC。例如，如果一个 VNIC 与三个 MAC 地址关联，则会为虚拟网络驱动程序指定其中一个 MAC 地址。因此，您只能使用其余两个 MAC 地址创建两个 VNIC。

可以使用下面的命令显示与 VNIC 关联的多个 MAC 地址：

```
# dladm show-vnic -m
```

例 2-9 显示内核区域中具有多个 MAC 地址的 VNIC

```
# dladm show-vnic -m
LINK          OVER      MACADDRESSES  MACADDRTYPES  VIDS
gz_vnic0      net0      2:8:20:d7:27:9d  random         0
zone1/net0    net0      2:8:20:70:52:9   random         0
              2:8:20:c9:d:4c   fixed
              2:8:20:70:db:3   random
zone1/net1    net0      0:1:2:3:4:5     fixed          0
              0:1:2:3:4:6     fixed
```

在此示例中，内核区域 zone1 有两个 anet 资源：net0 和 net1。这两个资源都配置有多个 MAC 地址。因此，在内核区域 zone1 内，您在与数据链路 net0 关联的虚拟 NIC 驱动程序 zvnet 之上最多可以创建两个 VNIC。您只能在与数据链路 net1 关联的虚拟 NIC 驱动程序 zvnet 之上创建一个 VNIC。

例 2-10 显示具有多个 MAC 地址的由系统创建的 VNIC

```
# dladm show-vnic -m
LINK          OVER      MACADDRESSES  MACADDRTYPES  VIDS
ldoms-vsw0.vport0 net1      0:14:4f:fb:e1:8f  fixed          0,21
              0:14:4f:f8:6b:9   fixed
              0:14:4f:fa:48:7f  fixed
ldoms-vsw0.vport1 net1      0:14:4f:f9:1b:8d  fixed          45,44
              0:14:4f:f9:27:4   fixed
```

在此示例中，您在与 ldoms-vsw0.vport0 关联的来宾域的虚拟网络驱动程序 vnet 之上最多可以创建两个 VNIC。您在与 ldoms-vsw0.vport1 关联的虚拟 NIC 驱动程序 vnet 之上最多可以创建一个 VNIC。

显示数据链路的物理和虚拟链路状态

数据链路的物理链路状态指明物理设备是否连接到外部网络。如果插入了电缆，并且电缆另一端的端口的状态为 up，则表示物理设备连接到外部网络。

可以使用下面的命令显示数据链路的物理链路状态：

```
# dladm show-phys [link]
```

```
# dladm show-ether [link]
```

有关更多信息，请参见 [dladm\(1M\)](#) 手册页。

例 2-11 显示数据链路的物理链路状态

下面的示例通过使用 `dladm show-phys` 命令显示系统上的数据链路的物理链路状态。

```
# dladm show-phys
LINK      MEDIA      STATE      SPEED  DUPLEX    DEVICE
net1      Ethernet   down       0      unknown  e1000g1
net2      Ethernet   down       0      unknown  e1000g2
net3      Ethernet   down       0      unknown  e1000g3
net0      Ethernet   up         1000   full     e1000g0
```

下面的示例通过使用 `dladm show-ether` 命令显示系统上的数据链路的物理链路状态。

```
# dladm show-ether
LINK      PTYPE     STATE     AUTO  SPEED-DUPLEX  PAUSE
net1      current   down     yes   0M             bi
net2      current   down     yes   0M             bi
net3      current   down     yes   0M             bi
net0      current   up       yes   1G-f           bi
```

当基于一个 NIC 创建了多个 VNIC 时，会在内部创建一个虚拟交换机，以便使处于同一 VLAN 上的 VNIC 和主数据链路能够通信。这些数据链路可以相互通信，即使物理数据链路没有与外部网络连接也是如此。这构成了数据链路的虚拟链路状态，该状态可以是 up、down 或 unknown。数据链路的虚拟链路状态指明数据链路是否连接到系统内的内部网络（即使未插入物理电缆）。

可以使用下面的命令显示数据链路的虚拟链路状态：

```
# dladm show-link [link]
```

例 2-12 显示数据链路的虚拟链路状态

此示例显示了系统上的数据链路的虚拟链路状态。

```
# dladm show-link
LINK      CLASS     MTU      STATE  OVER
net0      phys     1500    up     --
net2      phys     1500    down   --
net4      phys     1500    down   --
net1      phys     1500    up     --
net5      phys     1500    up     --
vnic0     vnic     1500    up     net5
vnic1     vnic     1500    up     net5
vnic2     vnic     1500    up     net1
```

修改 VNIC 的 VLAN ID

VNIC 可配置为 VLAN。当您希望 VNIC 托管特定 VLAN 的通信时，需要修改数据链路上的 VNIC 的 VLAN ID。

使用的 `dladm` 子命令取决于是要修改 VLAN 还是修改配置为 VLAN 的 VNIC：

- 对于使用 `dladm create-vlan` 命令创建的 VLAN，请使用 `dladm modify-vlan` 命令。要显示这些 VLAN，请使用 `dladm show-vlan` 命令。
- 对于使用 `dladm create-vnic` 命令创建的 VLAN，请使用 `dladm modify-vnic` 命令。要显示这些 VNIC（包括那些带有 VLAN ID 的 VNIC），请使用 `dladm show-vnic` 命令。

可以修改在数据链路上配置的单个 VNIC 或多个 VNIC 的 VLAN ID。还可以通过为所有 VNIC 配置同一个 VLAN ID，以组的形式修改 VNIC 的 VLAN ID。

- 如果只在数据链路上配置了一个 VNIC，则使用下面的命令语法修改该 VNIC 的 VLAN ID：

```
# dladm modify-vnic -v vid -L link
```

其中，*vid* 是为 VNIC 指定的 VLAN ID。

例 2-13 修改数据链路上的一个 VNIC 的 VLAN ID

在此示例中，修改了在数据链路 `net0` 上配置的 `vnic0` 的 VLAN ID。

```
# dladm modify-vnic -v 123 -L net0
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net0      1000      2:8:20:c2:39:38  random           123
```

- 如果在数据链路上配置了多个 VNIC，则使用下面的命令语法修改这些 VNIC 的 VLAN ID：

```
# dladm modify-vnic -v vid VNIC
```

因为每个 VLAN ID 对于同一数据链路上的 VNIC 都是唯一的，所以，一次只能更改一个 VLAN ID。

例 2-14 修改数据链路上的多个 VNIC 的 VLAN ID

在此示例中，修改了 `vnic0`、`vnic1` 和 `vnic2` 的 VLAN ID。

```
# dladm modify-vnic -v 123 vnic0
# dladm modify-vnic -v 456 vnic1
# dladm modify-vnic -v 789 vnic2
# dladm show-vnic
```

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VIDS
vnic0	net0	1000	2:8:20:c2:39:38	random	123
vnic1	net0	1000	2:8:20:5f:84:ff	random	456
vnic2	net0	1000	2:8:20:5f:84:ff	random	789

- 如果在不同的数据链路上配置了每个 VNIC，则使用下面的命令语法以组形式修改 VNIC 的 VLAN ID：

```
# dladm modify-vnic -v vid VNIC,VNIC,[...]
```

例 2-15 以组形式修改 VNIC 的 VLAN ID

在此示例中，以组形式修改了 vnic0、vnic1 和 vnic2 的 VLAN ID。这些 VNIC 分别配置在数据链路 net0、net1 和 net2 上。

```
# dladm modify-vnic -v 123 vnic0,vnic1,vnic2
# dladm show-vnic
```

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VIDS
vnic0	net0	1000	2:8:20:c2:39:38	random	123
vnic1	net1	1000	2:8:20:5f:84:ff	random	123
vnic2	net2	1000	2:8:20:5f:84:ff	random	123

修改 VNIC MAC 地址

用户创建的任何 VNIC 都只能有一个 MAC 地址。可以使用 `dladm modify-vnic` 命令修改 MAC 地址。对于为内核区域创建的 VNIC，您可以为其配置一个或多个 MAC 地址。

可以修改在数据链路上配置的 VNIC 的现有 MAC 地址。可以修改所有 VNIC 的 MAC 地址，也可以有选择地修改指定 VNIC 的 MAC 地址。还可以同时修改 VNIC 的 VLAN ID 和 MAC 地址。

- 要修改 VNIC 的 MAC 地址，请使用下面的命令语法：

```
# dladm modify-vnic -m MAC-address VNIC
```

其中，*MAC-address* 是指要指定给 VNIC 的新 MAC 地址。

例 2-16 修改 VNIC 的 MAC 地址

在此示例中，为 vnic0 指定了一个特定的 MAC 地址。

```
# dladm modify-vnic -m 3:8:20:5f:84:ff vnic0
# dladm show-vnic
```

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VIDS
vnic0	net0	1000	3:8:20:5f:84:ff	fixed	0

- 要修改数据链路上的所有 VNIC 的 MAC 地址，请使用下面的命令语法：

```
# dladm modify-vnic -m random -L link
```

在此命令语法中，`-m random` 选项等效于 `-m auto` 选项。将自动地随机将 MAC 地址指定给 VNIC。

例 2-17 修改数据链路上的所有 VNIC 的 MAC 地址

在此示例中，将自动地随机修改在数据链路 `net0` 上配置的所有 VNIC 的 MAC 地址。

```
# dladm modify-vnic -m random -L net0
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net0      1000       2:8:20:22:9d:bb  random           0
vnic1     net0      1000       2:8:20:72:2e:9   random           0
vnic2     net0      1000       2:8:20:2f:e5:83  random           0
```

- 要有选择地修改 VNIC 的 MAC 地址，请使用下面的命令语法：

```
# dladm modify-vnic -m random VNIC,VNIC,[...]
```

对于全局修改和有选择的修改，您都为 `-m` 选项指定了 `random`。

例 2-18 有选择地修改 VNIC 的 MAC 地址

在此示例中，有选择地修改了在数据链路 `net0` 上配置的 `vnic0` 和 `vnic2` 的 MAC 地址。

```
# dladm modify-vnic -m random vnic0,vnic2
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net0      1000       2:8:20:2f:e5:83  random           0
vnic1     net0      1000       2:8:20:5f:84:ff  fixed            0
vnic2     net0      1000       2:8:20:2f:e5:83  random           0
```

- 要同时修改 VNIC 的 VLAN ID 和 MAC 地址，请使用下面的命令语法：

```
# dladm modify-vnic -m random -v vid VNIC
```



注意 - 全局修改 VNIC 的多个属性可能会导致出现意外的 VNIC 行为。因此，应分别修改 VNIC 的多个属性。

例 2-19 修改 VNIC 的 VLAN ID 和 MAC 地址

在此示例中，同时修改了 `vnic0` 的 VLAN ID 和 MAC 地址。

```
# dladm modify-vnic -m random -v 123 vnic0
# dladm show-vnic vnic0
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
```

```
vnic0    net0    1000    2:8:20:2f:e5:83    random    123
```

迁移 VNIC

您可以将一个或多个 VNIC 从一个底层数据链路移动到另一个底层数据链路，而无需删除和重新配置 VNIC。底层数据链路可以是物理链路、链路聚合或 etherstub。

通常在以下任意情况下迁移 VNIC：

- 当您需要将现有的 NIC 替换为新的 NIC 时
- 当目标 NIC 的带宽多于现有 NIC 的带宽时
- 当目标 NIC 在硬件中实现某些功能（例如，大型接收负载转移 (large receive offload, LRO)、大型段负载转移 (large segment offload, LSO) 和校验和）时

要成功迁移 VNIC，要将 VNIC 移动到的目标数据链路必须能够包容 VNIC 的数据链路属性。如果不支持这些属性，则迁移将会失败并通知用户。成功迁移后，如果目标数据链路连接到网络，则使用这些 VNIC 的所有应用程序将继续正常运行。

某些与硬件相关的属性在 VNIC 迁移后可能会发生更改，例如数据链路状态、链路速度和 MTU 大小。这些属性的值继承自向其迁移 VNIC 的数据链路。可以迁移在数据链路上配置的所有 VNIC，也可以有选择地迁移指定的 VNIC。还可以同时迁移 VNIC 并修改其 VLAN ID。

- 要将在源链路上配置的所有 VNIC 迁移到目标链路，请使用下面的命令语法：

```
# dladm modify-vnic -l target-link -L source-link
```

-l *target-link* 指迁移后的 VNIC 所基于的链路

-L *source-link* 指 VNIC 以前配置在其上的链路

例 2-20 将所有 VNIC 从源链路迁移到目标链路

在此示例中，源链路 ether0 中的所有 VNIC 将移到目标链路 net1。

```
# dladm modify-vnic -l net1 -L ether0
```

```
# dladm show-vnic
```

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VIDS
vnic0	net1	1000	2:8:20:c2:39:38	random	321
vnic1	net1	1000	2:8:20:5f:84:ff	random	656
vnic2	net1	1000	2:8:20:5f:84:ff	random	0

- 要将在源链路上配置的指定 VNIC 迁移到目标链路，请使用下面的命令语法：

```
# dladm modify-vnic -l target-link VNIC,VNIC,[...]
```

要执行选择性 VNIC 迁移，您需要仅指定目标链路。

例 2-21 将指定的 VNIC 从源链路迁移到目标链路

在此示例中，有选择地将 vnic0、vnic1 和 vnic2 从源链路 net0 移动到目标链路 net1。

```
# dladm modify-vnic -l net1 vnic0,vnic1,vnic2
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net1      1000      2:8:20:c2:39:38  random           321
vnic1     net1      1000      2:8:20:5f:84:ff  random           656
vnic2     net1      1000      2:8:20:5f:84:ff  random           0
vnic3     net0      1000      2:8:20:5f:84:ff  random           345
```

- 要同时修改在源链路上配置的 VNIC 的 VLAN ID 并将这些 VNIC 迁移到目标链路，请使用下面的命令语法：

```
# dladm modify-vnic -l target-link -v vid VNIC
```

要指定新的 VLAN ID，必须一次迁移一个 VNIC。

例 2-22 迁移并修改 VNIC 的 VLAN ID

在此示例中，将 vnic0、vnic1 和 vnic2 迁移到目标数据链路 net1。迁移的同时，还会修改所有 VNIC 的 VLAN ID。

```
# dladm modify-vnic -l net1 -v 123 vnic0
# dladm modify-vnic -l net1 -v 456 vnic1
# dladm modify-vnic -l net1 -v 789 vnic2
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net1      1000      2:8:20:c2:39:38  random           123
vnic1     net1      1000      2:8:20:5f:84:ff  random           456
vnic2     net1      1000      2:8:20:5f:84:ff  random           789
```

将 VNIC 从源链路迁移到目标链路时，随机指定的 MAC 地址在迁移后不受影响并由其各自的 VNIC 保留。请参见例 2-22 “迁移并修改 VNIC 的 VLAN ID”。

但是，如果 VNIC 使用的是源链路的出厂 MAC 地址，则 MAC 地址将发生更改。如果迁移过程中未指定 MAC 地址，则 VNIC 的出厂 MAC 地址将替换为随机指定的 MAC 地址。如果在迁移过程中使用 -m 指定了 MAC 地址，则 VNIC 的出厂 MAC 地址将替换为指定的 MAC 地址。

您具有多个与内核区域创建的 VNIC 关联的 MAC 地址。当迁移内核区域创建的 VNIC 时，与 VNIC 关联的所有 MAC 地址都会迁移到目标 NIC。

删除 VNIC

本节介绍了如何删除 VNIC。

▼ 如何删除 VNIC

1. 成为管理员。
2. (可选) 检查 VNIC 是否繁忙。

仅当 VNIC 不忙时，您才能删除它。VNIC 可能出于多个原因而繁忙。您需要执行下面的步骤来检查 VNIC 是否繁忙：

- 检查 VNIC 是否已激活且与一个 IP 地址相关联。

```
# ipadm show-if  
# ipadm show-addr
```

如果 VNIC 已激活且与一个 IP 地址相关联，则删除 IP 接口。

```
# ipadm delete-ip interface
```

- 检查在 VNIC 上是否配置有任何流。

```
# flowadm
```

如果在 VNIC 上配置了流，则删除流。

```
# flowadm remove-flow flowname
```

- 检查是否将 VNIC 指定给了区域。

```
# dladm show-link -Z
```

有关如何删除连接到区域的 VNIC 的更多信息，请参见[如何删除连接到区域的 VNIC \[44\]](#)。

- 检查 VNIC 是否是由系统创建的。

```
# dladm show-vnic
```

只有系统创建的 VNIC 会包含连字符 (-)，这可帮助您区分系统创建的 VNIC 和用户创建的 VNIC。无法修改、重命名、激活或删除系统创建的 VNIC。

- 检查是否已侦测到 VNIC。

```
# snoop  
# tshark
```

如果已使用 snoop 命令侦测到 VNIC，则终止该进程。

```
# pkill snoop
```

如果已使用 tshark 命令侦测到 VNIC，则终止该进程。

```
# pkill tshark
```

3. 删除 VNIC。

```
# dladm delete-vnic VNIC
```

▼ 如何删除连接到区域的 VNIC

此过程假设 VNIC 已连接到区域。您必须在全局区域中执行本过程。

1. 停止区域。

```
global# zoneadm -z ZONE halt
```

注 - 要确定区域所使用的链路，请使用 `dladm show-link` 命令。

2. 从区域中删除或分离 VNIC。

```
global# zonecfg -z ZONE remove net physical=VNIC
```

3. 从系统中删除 VNIC。

```
global# dladm delete-vnic VNIC
```

4. 重新引导区域。

```
global# zoneadm -z ZONE boot
```

例 2-23 删除连接到区域的 VNIC

在本示例中，将从 zoneB 和系统中删除 vnic1。

```
global# dladm show-link
LINK          CLASS  MTU   STATE  OVER
net0          phys  1500  up     --
net2          phys  1500  up     --
net1          phys  1500  up     --
net3          phys  1500  up     --
zoneA/net0    vnic   1500  up     net0
zoneB/net0    vnic   1500  up     net0
vnic0         vnic   1500  up     net1
zoneA/vnic0   vnic   1500  up     net1
vnic1         vnic   1500  up     net1
zoneB/vnic1   vnic   1500  up     net1

global# zoneadm -z zoneB halt
global# zonecfg -z zoneB remove net physical=vnic1
```

```
global# dladm delete-vnic vnic1
global# zoneadm -z zoneB boot
```

将单根 I/O 虚拟化与 VNIC 一起使用

从 Oracle Solaris 11.2 发行版开始，您可以使用 `dladm` 命令管理支持单根 I/O 虚拟化 (single root I/O virtualization, SR-IOV) 的网络设备。SR-IOV 是一项标准，允许在虚拟机之间高效共享外设部件高速互连 (Peripheral Component Interconnect Express, PCIe) 设备。它是在硬件中实现的。有关更多信息，请参见《编写适用于 Oracle Solaris 11.2 的设备驱动程序》中的“SR-IOV 简介”。

启用数据链路的 SR-IOV 模式

在 Oracle Solaris 中，可以将支持 SR-IOV 的网络设备的虚拟功能 (VF) 与 VNIC 或 VLAN 关联。VF VNIC 是拥有专用 VF 的 VNIC。在资源的共享方面，VF VNIC 不同于常规 VNIC。常规 VNIC 需要与其他常规 VNIC 共享资源，但 VF VNIC 不需要共享资源。每个 VF 都是 VF VNIC 的单独硬件资源。

只能在支持 SR-IOV 模式的数据链路上创建 VF VNIC。缺省情况下，数据链路的 SR-IOV 模式被禁用。通过将 `iovs` 属性设置为 `on`，可以启用数据链路的 SR-IOV 模式。有关在启用数据链路的 SR-IOV 模式后创建 VF VNIC 的信息，请参见“创建 VF VNIC” [46]。

通过随 `dladm show-linkprop` 命令指定链路属性 `iovs`，可以检查数据链路的 SR-IOV 模式。如果输出的 `EFFECTIVE` 列下的值为 `off`，则表示数据链路的 SR-IOV 模式处于禁用状态。

以下示例说明了如何检查数据链路 `net0` 的 SR-IOV 模式。

```
# dladm show-linkprop -p iovs net0
LINK      PROPERTY  PERM    VALUE    EFFECTIVE  DEFAULT  POSSIBLE
net0     iovs      rw      auto     off        auto     auto,on,off
```

在此示例中，数据链路 `net0` 的 SR-IOV 模式处于禁用状态。该输出显示了以下信息：

VALUE 指定您已为 `iovs` 链路属性设置的值。如果您尚未修改 `iovs` 链路属性，则 `iovs` 链路属性的缺省值为 `auto`。值为 `auto` 意味着将由操作系统来确定在缺省情况下是否在特定的物理数据链路上启用 SR-IOV 模式。

EFFECTIVE 数据链路的实际 SR-IOV 模式。缺省情况下，所有支持 SRIOV 的 NIC 在 `EFFECTIVE` 列下都显示值 `off`。

通过将 `iovs` 属性设置为 `on`，可以启用数据链路 `net0` 的 SR-IOV 模式，如下所示：

```
# dladm set-linkprop -p iov=on net0
# dladm show-linkprop -p iov net0
LINK      PROPERTY  PERM    VALUE    EFFECTIVE  DEFAULT  POSSIBLE
net0     iov       rw      on       on         auto     auto,on,off
```

同样，可以通过将 `iovs` 链路属性设置为 `off` 来禁用数据链路的 SR-IOV 模式。有关 `dladm` 命令的更多信息，请参见 [dladm\(1M\)](#) 手册页。

创建 VF VNIC

要在数据链路上创建 VF VNIC，您需要启用数据链路的 SR-IOV 模式。有关更多信息，请参见“[启用数据链路的 SR-IOV 模式](#)” [45]。启用数据链路的 SR-IOV 模式后，当您使用 `dladm create-vnic` 命令创建 VNIC 时，会自动将 VF 分配给 VNIC。同样，在您使用 `dladm create-vlan` 命令创建 VLAN 时，会自动将 VF 分配给 VLAN。

还可以通过随 `dladm create-vnic` 或 `dladm create-vlan` 命令指定 `iovs` VNIC 链路属性来显式指定是否需要将 VF 分配给 VNIC 或 VLAN。

可以使用下面的命令语法显式创建 VF VNIC：

```
# dladm create-vnic [-p iov=value] -l link VNIC
```

创建 VF VNIC 时，是否指定 `iovs` VNIC 链路属性是可选的。如果未指定 `iovs` VNIC 链路属性，则会将缺省值 `inherit` 指定给此属性。可以为 `iovs` VNIC 链路属性指定以下值：

<code>inherit</code>	<code>iovs</code> VNIC 链路属性的缺省值。确定是否需要基于底层数据链路的有效 <code>iovs</code> 属性值分配 VF： <ul style="list-style-type: none"> ▪ <code>off</code> – 不为 VNIC 分配 VF。 ▪ <code>on</code> – 尝试为 VNIC 分配 VF。如果无法分配，则创建常规 VNIC。
<code>on</code>	分配 VF。如果找不到 VF，则 VNIC 创建将失败。
<code>off</code>	创建不带 VF 的 VNIC。

对数据链路使用 `dladm show-linkprop` 命令时，数据链路属性的生效值是在 `EFFECTIVE` 列下显示的值。

`iovs` VNIC 链路属性与其他数据链路属性之间的区别在于，只有在创建 VNIC 或 VLAN 时才能指定 `iovs` VNIC 链路属性。在创建 VNIC 或 VLAN 后，无法修改 `iovs` VNIC 链路属性。

ioV VNIC 链路属性具有一个生效值，该值指示是否为 VNIC 或 VLAN 分配 VF。EFFECTIVE 列下的值 on 表示已分配 VF，EFFECTIVE 列下的值 off 表示未分配 VF。

例 2-24 创建 VF VNIC

以下示例说明了如何通过显式指定 ioV VNIC 链路属性，在数据链路 net0 上创建 VF VNIC vfvnic1 和常规 VNIC vnic1。此示例假设您已启用数据链路 net0 的 SR-IOV 模式。

```
# dladm show-linkprop -p iov net0
LINK      PROPERTY PERM   VALUE EFFECTIVE DEFAULT POSSIBLE
net0     iov      rw    on    on      auto   auto,on,off
# dladm create-vnic -l net0 vfvnic1
# dladm show-linkprop -p iov vfvnic1
LINK      PROPERTY PERM   VALUE EFFECTIVE DEFAULT POSSIBLE
vfvnic1  iov      r-    inherit on      inherit inherit,on,off
# dladm create-vnic -p iov=off -l net0 vnic1
# dladm show-linkprop -p iov vnic1
LINK      PROPERTY PERM   VALUE EFFECTIVE DEFAULT POSSIBLE
vnic1    iov      r-    off   off     inherit inherit,on,off
```

此示例提供了以下信息：

- 在创建 VF VNIC 之前，您需要将数据链路 net0 的 iov 属性设置为 on。
- 如果在创建 VNIC 时未指定 iov 属性的值，则会将缺省值 inherit 指定给 iov 属性。创建带有 VF 的 VF VNIC vfvnic1。
- 如果在创建 VNIC 时显式为 iov 属性指定了值 off，则将创建不带 VF 的常规 VNIC，即使底层数据链路 net0 的 iov 属性为 on 也是如此。创建不带 VF 的 VNIC vnic1。

迁移 VF VNIC

可以将 VF VNIC 或 VF VLAN 从一个数据链路移到另一个数据链路。请注意以下要求：

- 目标数据链路必须支持 SR-IOV，并且 iov 属性必须设置为 on。有关如何检查数据链路的 iov 属性状态的更多信息，请参见“[启用数据链路的 SR-IOV 模式](#)” [45]。
- VF 必须在目标数据链路上可用。有关如何检查数据链路上可用 VF 的数量的更多信息，请参见“[显示 VF 信息](#)” [48]。

如果不满足这些要求，则会将 VF VNIC 作为不带 VF 的常规 VNIC 迁移到目标数据链路。

如果迁移通过指定 iov=inherit 创建的 VF VNIC，则即使目标数据链路不支持 iov 属性或 iov 属性已禁用，迁移也会成功。如果尝试迁移使用 iov=on 创建的 VF VNIC，则只有目标数据链路上启用了 SR-IOV 模式时，迁移才会成功。

有关如何迁移 VNIC 的更多信息，请参见“[迁移 VNIC](#)” [41]。

显示 VF 信息

可以使用下面的命令显示有关数据链路上 VF 的可用性的信息：

```
# dladm show-phys -V
```

该输出显示了以下信息：

LINK	数据链路的名称。
VFS-AVAIL	数据链路上可指定给 VNIC 的可用 VF 数量。如果数据链路不支持 SR-IOV，则 VFS-AVAIL 显示为 --。
VFS-INUSE	数据链路所使用的 VF 数量。如果数据链路不支持 SR-IOV，则 VFS-INUSE 显示为 --。
FLAGS	l 标志指示数据链路由 Oracle VM Server for SPARC 管理。

例 2-25 显示数据链路的 VF 信息

```
# dladm show-phys -V
LINK      VFS-AVAIL  VFS-INUSE  FLAGS
net0      30         1          -----
net1      0          0          l-----
net2      --         --         -----
```

在此示例中，数据链路 net0 具有 30 个可用的 VF，一个 VF 正在使用中。数据链路 net1 没有 (0) 可用的 VF，并且当前正在由 Oracle VM Server for SPARC 使用。数据链路 net2 不支持 SR-IOV。

可以使用下面的命令显示指定给系统上的 VNIC 的 VF 设备：

```
# dladm show-vnic -V
```

该输出显示了以下信息：

LINK	VNIC 的名称。
VF-ASSIGNED	指定给 VNIC 的 VF 设备。如果 VNIC 没有 VF，则 VF-ASSIGNED 显示为 --。

例 2-26 显示指定给 VNIC 的 VF 设备

```
# dladm show-vnic -V
```

LINK	VF-ASSIGNED
vnic1	ixgbev0
vnic2	--
vnic3	ixgbev1

在此示例中，VF 设备 `ixgbev0` 指定给了 `vnic1`。没有为 VNIC `vnic2` 指定 VF 设备。VF 设备 `ixgbev1` 指定给了 `vnic3`。

使用虚拟可扩展局域网配置虚拟网络

诸如虚拟局域网 (virtual local area network, VLAN) 这样的传统网络隔离方法不足以支持大型数据中心的虚拟化。因为云环境也与底层物理网络紧密结合，所以不能在属于不同的第 2 层物理网络的物理服务器之间迁移虚拟机。Oracle Solaris 支持虚拟可扩展局域网 (virtual extensible local area network, VXLAN) 技术，此技术用于解决大型虚拟化数据中心或云环境中的这类虚拟化问题。

本章概述了如何部署 VXLAN 并介绍了如何配置它们。其中还介绍了如何将 VXLAN 与其他技术（例如，区域）一起使用。

本章包含以下主题：

- [“VXLAN 概述” \[51\]](#)
- [“使用 VXLAN 的优点” \[52\]](#)
- [“VXLAN 命名约定” \[52\]](#)
- [“VXLAN 拓扑” \[53\]](#)
- [“结合使用 VXLAN 和区域” \[55\]](#)
- [“配置 VXLAN” \[57\]](#)
- [“显示 VXLAN 信息” \[61\]](#)
- [“删除 VXLAN” \[62\]](#)
- [“将 VXLAN 分配给区域” \[62\]](#)
- [“使用案例：在链路聚合上配置 VXLAN” \[63\]](#)

VXLAN 概述

在云环境中，物理服务器可能位于不同的第 2 层网络中。例如，云可能跨越位于不同地理位置的多个物理服务器。在这种情况下，在第 2 层网络上创建虚拟机 (virtual machine, VM) 或租户将限制可用于置备这些 VM 的物理服务器的数量。您可以使用不同的第 2 层网络中的物理服务器来置备 VM。然而，由于不同服务器之间的迁移限制于相同的第 2 层网络，因此物理资源的利用率并未得到优化。

VXLAN 是第 2 层技术，使用此技术可以在第 3 层网络之上创建第 2 层网络，从而提供进一步的网络隔离。VXLAN 提供一个伸展横跨多个第 2 层物理网络的第 2 层虚拟网

络。因此，在云环境中置备资源并不限于单个第 2 层物理网络。物理服务器可以作为 VXLAN 网络的一部分，只要通过 IPv4 或 IPv6 网络连接这些服务器即可。

可以结合使用 VXLAN 技术和 Oracle Solaris 的弹性虚拟交换机 (Elastic Virtual Switch, EVS) 功能来创建大量的虚拟网络。有关如何结合使用 VXLAN 和 EVS 功能创建虚拟网络的信息，请参见“[使用案例：为租户配置弹性虚拟交换机](#)” [136]。有关更多信息，请参见第 5 章 [关于弹性虚拟交换机](#) 和第 6 章 [管理弹性虚拟交换机](#)。

VXLAN 提供隔离的第 2 层网段，该网段通过 VXLAN 网段 ID 或 VXLAN 网络标识符 (VNI) 进行标识。同一 VXLAN 网段中的所有 VM 属于同一个第 2 层虚拟广播域。

VXLAN 中的通信类似于隔离 VLAN 中的通信。因此，只有在同一个 VXLAN 网段中的 VM 才能相互通信。不在同一个 VXLAN 网段中的 VM 不能相互通信。

使用 VXLAN 的优点

VXLAN 具有以下优点：

- 因为 VXLAN ID 是 24 位的，所以可提高虚拟化云环境中的可扩展性，使您能够创建多达 1600 万个隔离网络。这克服了具有 12 位 VLAN ID 的 VLAN 的限制，VLAN 使您能够创建多达 4094 个隔离网络。
- 使您能够使用底层网络的第 3 层功能。
- 第 2 层虚拟网络是从底层物理网络抽象出来的。因此，该虚拟网络对物理网络不可见并且具备以下优点：
 - 无需使用其他物理基础结构。例如，外部交换机的转发表不会随着服务器上物理端口后的 VM 数目增加而增长。
 - 缩小位于同一 VXLAN 网段中的 VM 的 MAC 地址重复范围。当 MAC 地址不属于同一个 VXLAN 网段时，这些地址可以重叠。

在 VXLAN 中，仅属于同一个 VXLAN 网段或 VNI 的数据链路的 MAC 地址必须唯一。这类似于 VLAN ID 和 MAC 地址必须具有唯一组合的 VLAN。

VXLAN 命名约定

在 Oracle Solaris 中，VXLAN 端点通过 VXLAN 数据链路表示。此 VXLAN 数据链路 with IP 地址 (IPv4 或 IPv6) 和 VXLAN 网络标识符 (VNI) 相关联。尽管多个 VXLAN 数据链路可以使用同一个 IP 地址，但是 IP 地址和 VNI 的组合必须唯一。可以为 VXLAN 数据链路配置可选的多播地址，此地址用于在同一个 VNI 上发现对等 VXLAN 端点，并且还实现 VXLAN 网段内的广播。必须为同一个 VNI 中的 VXLAN 数据链路配置相同的多播地址。有关 VXLAN 要求的更多信息，请参见“[VXLAN 要求](#)” [57]。

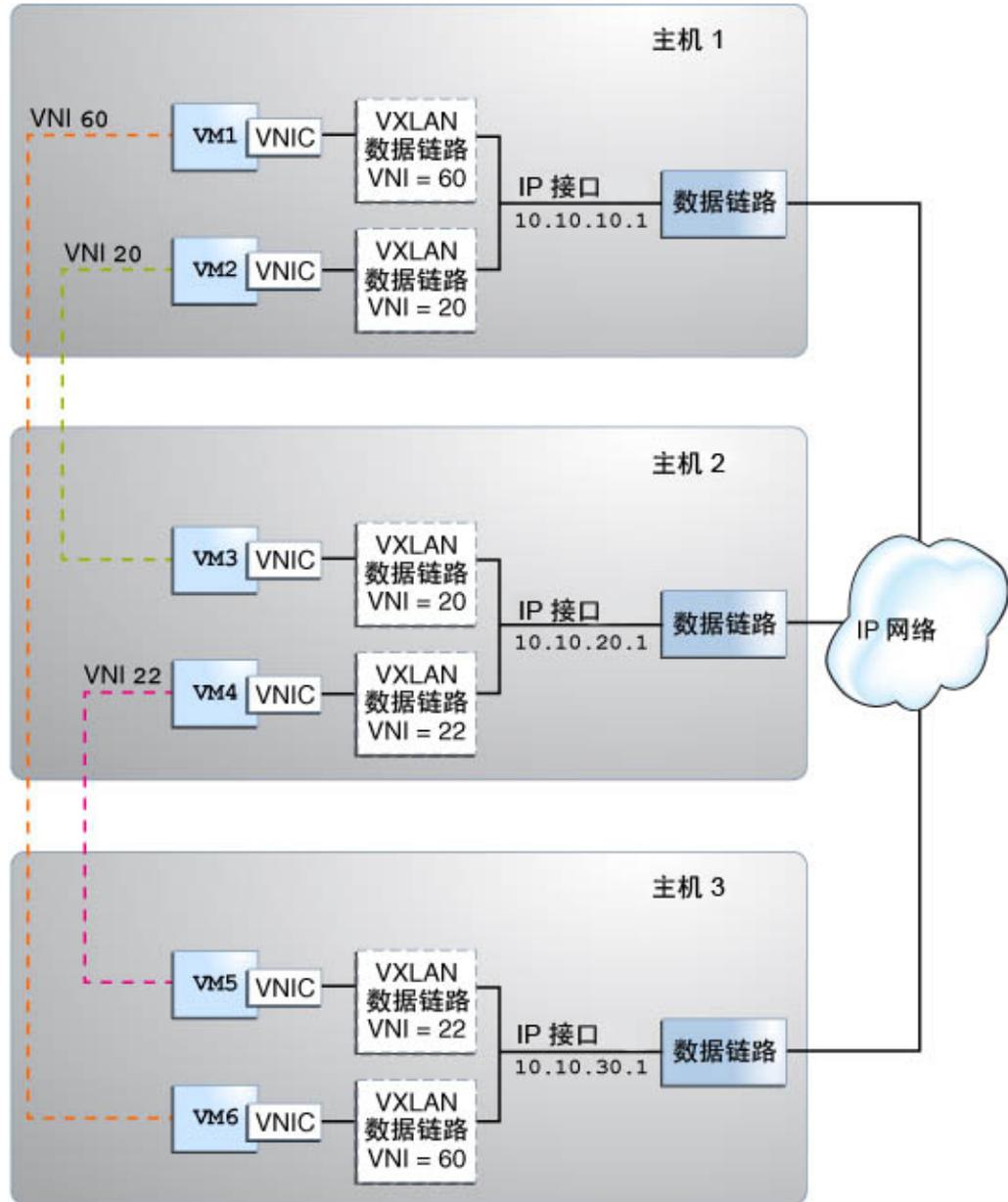
每个 VXLAN 数据链路均与一个 VXLAN 网段 ID 或 VNI 相关联。VXLAN 数据链路的命名约定与用于链路或 VLAN 的约定相同。有关提供有效数据链路名称的信息，请参见《[在 Oracle Solaris 11.2 中配置和管理网络组件](#)》中的“有效链路名称的规则”。

VXLAN 拓扑

使用 VXLAN 可以在第 3 层网络上将系统组织到其自己的 VXLAN 网段内。

下图说明了在多个物理服务器上配置的 VXLAN 网络。

图 3-1 VXLAN 拓扑



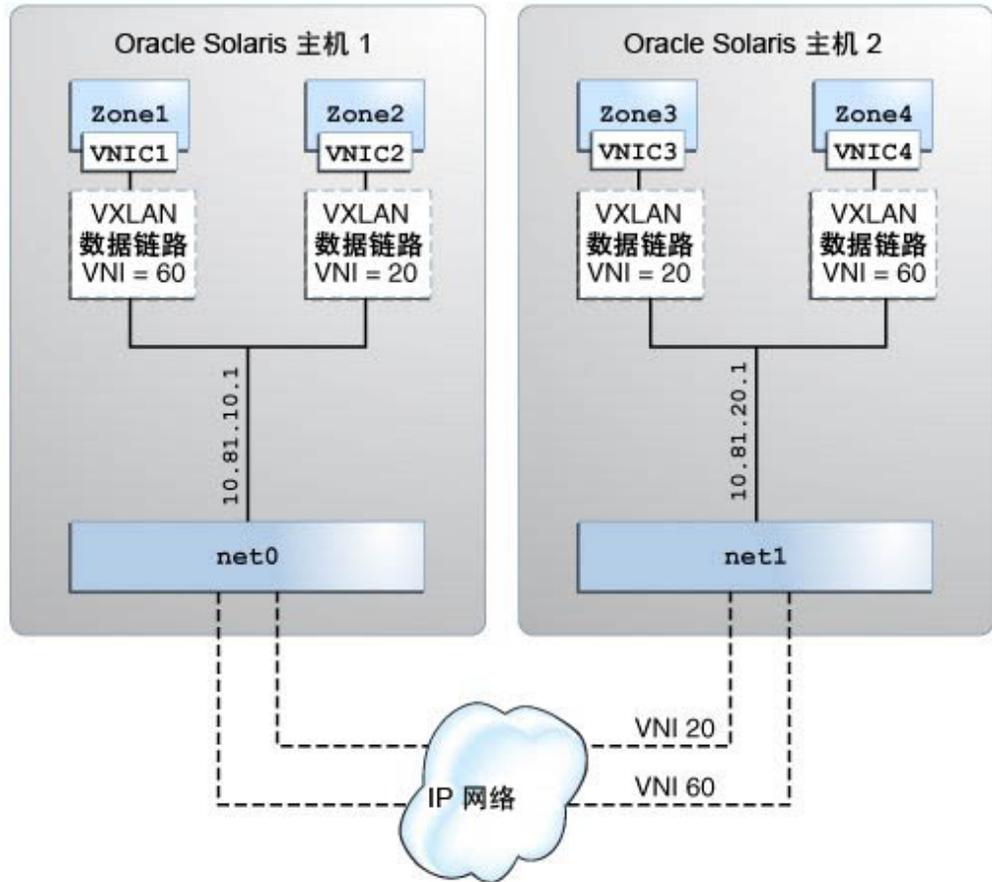
此图显示了三个连接到 IP 网络基础结构的虚拟化主机。存在三个由 VXLAN 网段 ID 或 VNI 标识的 VXLAN 覆盖网络：60、20 和 22。VM VM1 和 VM6 位于由 VNI 60 标识的覆盖网络上，VM VM2 和 VM3 位于由 VNI 20 标识的覆盖网络上，VM VM4 和 VM5 位于由 VNI 22 标识的覆盖网络上。

结合使用 VXLAN 和区域

可以将 VXLAN 数据链路上创建的 VNIC 分配给区域。VXLAN 数据链路是通过指定 VNI 创建的，并且这些 VXLAN 数据链路属于由该 VNI 标识的 VXLAN 网段。例如，如果在创建 VXLAN 数据链路时将 VNI 指定为 20，则该数据链路将属于由 VNI 20 标识的 VXLAN 网段。在 VXLAN 数据链路上创建的 VNIC 属于 VXLAN 网段。

下图显示了两个连接到 IP 网络基础结构的虚拟化 Oracle Solaris 主机，和两个由 VNI 20 和 60 标识的 VXLAN 覆盖网络。

图 3-2 VXLAN 和区域



可以通过以下方式创建属于 VXLAN 网段的区域：

- 在 VXLAN 上创建一个 VNIC 并将该 VNIC 分配给区域。有关更多信息，请参见“[配置 VXLAN](#)” [57]。
- 分配 VXLAN 作为区域的 anet (VNIC) 资源的底层链路。有关更多信息，请参见“[将 VXLAN 分配给区域](#)” [62]。

在任何情况下，在区域中创建的 VNIC 都属于由底层 VXLAN 数据链路标识的 VXLAN 网段。有关区域的更多信息，请参见 [《Oracle Solaris 11.2 虚拟环境介绍》](#)。

将 VNIC 分配给 VXLAN 链路类似于，创建一个 VLAN 链路并将其分配给区域。有关创建 VLAN 并将其分配给区域的更多信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的“如何配置 VLAN”。

规划 VXLAN 配置

规划 VXLAN 配置包括以下步骤：

1. 确定物理网络中的虚拟网络拓扑。例如，如果要托管的服务包含不同服务器上的若干个 VM，则可以为这些 VM 分配一个 VXLAN 网段。此 VXLAN 网段中的 VM 可以相互通信，但不能与不在此 VXLAN 网段中的其他 VM 通信。
2. 确认已通过 IP 接口连接物理服务器并且已在物理网络上启用 IP 多播。
3. 为 VXLAN 网段创建编号方案。例如，您可以根据 VM 托管的应用程序分配 VXLAN 网段 (VNI)。
4. 通过指定 IP 地址和 VXLAN 网段 ID 来创建 VXLAN 数据链路。
您可以有选择地为 VXLAN 网段分配其自己的多播地址。
5. 在 VXLAN 数据链路上创建 VNIC 并将 VNIC 分配给区域。
或者，您也可以分配 VXLAN 链路作为区域的 anet 链路的底层链路。

VXLAN 要求

在使用 VXLAN 之前，请检查您是否已满足以下要求：

- 确保网络上支持 IP 多播。如果不支持 IP 多播，则 VXLAN 中的 VM 不能相互通信。
- 如果 VXLAN 包含不同 IP 子网中的服务器，则这些子网中必须支持多播路由。如果不支持多播路由，则只有相同 IP 子网中的 VXLAN 上的 VM 才能相互通信，不同 IP 子网（例如，地理位置分散的数据中心）中的 VXLAN 上的 VM 不能相互通信。
有关 VXLAN 数据链路命名约定的更多信息，请参见“VXLAN 命名约定” [52]。

配置 VXLAN

以下过程假定已在系统上创建了区域。有关区域配置的信息，请参见《创建和使用 Oracle Solaris 区域》中的第 1 章“如何规划和配置非全局区域”。

▼ 如何配置 VXLAN

1. 成为管理员。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 确定系统上可用的 IP 地址。

```
# ipadm show-addr
```

3. 通过指定 IP 地址或 IP 接口创建 VXLAN 数据链路。

- 要通过指定 IP 地址创建 VXLAN，请执行以下操作：

```
# dladm create-vxlan -p prop=value VXLAN-LINK
```

`-p prop=value` 指定可在您创建的 VXLAN 数据链路上设置为指定值的 VXLAN 数据链路属性的逗号分隔列表。您可以设置以下属性：

- `addr` – 为 VXLAN 网络指定 IPv4 或 IPv6 地址。此地址可以是一个特定地址或地址/前缀长度的组合。
- `vni` – 指定 VXLAN 网段的网络标识符。可以指定一个介于 0 和 16777215 之间的数字。
- `mgroup` – (可选) 指定多播组名。仅当 VXLAN 网段具有其自己的多播组时，才能指定此选项。

`VXLAN-LINK` VXLAN 的名称。

- 要通过指定 IP 接口创建 VXLAN，请执行以下操作：

```
# dladm create-vxlan -p prop=value
```

`-p prop=value` 指定可在您创建的 VXLAN 数据链路上设置为指定值的 VXLAN 数据链路属性的逗号分隔列表。您可以设置以下属性：

- `interface` – 为 VXLAN 网络指定 IP 接口。
- `vni` – 指定 VXLAN 网段的网络标识符。可以指定一个介于 0 和 16777215 之间的数字。

`VXLAN` VXLAN 的名称。

当指定 IP 接口和 IP 版本时，将在该接口上指定的版本的可用 IP 地址上创建 VXLAN 数据链路。例如，如果在 `net0` 上配置了 IP 地址 `10.10.10.1`，则会在 `10.10.10.1` 上创建 VXLAN 数据链路。缺省情况下，IP 版本是 IPv4 地址。然而，如果您需要 IPv6 地址，必须通过使用 `ipvers` 属性来指定该版本。

注 - 可以在托管于物理聚合链路（中继聚合或 DLMP 聚合）或 IPoIB 链路的 IP 地址上创建 VXLAN 数据链路。但是，无法在托管于 IPMP、虚拟网络接口或回送接口的 IP 地址上创建 VXLAN 数据链路。

4. 验证所创建的 VXLAN。

```
# dladm show-vxlan
```

5. 在 VXLAN 数据链路上创建 VNIC。

```
# dladm create-vnic -l VXLAN-LINK VNIC
```

可以在 VXLAN 数据链路上创建 VLAN VNIC。要创建 VLAN VNIC，必须指定 `-f`（强制）选项。有关信息，请参见[如何配置带有 VLAN ID 的 VNIC \[24\]](#)。

6. 直接在 VNIC 上配置 IP 接口，或者通过首先将 VNIC 分配给区域进行配置。

- 在 VNIC 上配置 IP 接口。

```
# ipadm create-ip VNIC
```

```
# ipadm create-addr -a address VNIC
```

- 将 VNIC 分配给某个区域，然后在该区域内的 VNIC 上配置 IP 接口。

- a. 为 VNIC 分配该区域的接口。

```
zonecfg:zone> add net
zonecfg:zone:net> set physical=VNIC
zonecfg:zone:net> end
```

- b. 验证并提交您已实现的更改，然后退出该区域。

```
zonecfg:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

- c. 重新引导区域。

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

- d. 登录到区域。

```
global# zlogin zone
```

- e. 在该区域中，通过现已指定给该区域的 VNIC 创建 IP 接口。

```
zone# ipadm create-ip interface
```

- f. 使用有效的 IP 地址配置 VNIC。

如果您要为 VNIC 指定一个静态地址，需要键入以下内容：

```
zone# ipadm create-addr -a address interface
```

`-a address` 指定 IP 地址，该地址可以采用 CIDR 表示法。

g. 退出区域。

有关 `dladm` 和 `ipadm` 命令的信息，请参见 [dladm\(1M\)](#) 和 [ipadm\(1M\)](#) 手册页。

例 3-1 创建 VXLAN 并为在 VXLAN 上创建的 VNIC 配置 IP 接口

1. 检查系统上的可用 IP 地址。

```
# ipadm show-addr net4
ADDROBJ  TYPE  STATE  ADDR
net4/v4  static ok    10.10.11.1/24
```

2. 在 VXLAN 网段 10 中创建 VXLAN 数据链路。

```
# dladm create-vxlan -p addr=10.10.11.1,vni=10 vxlan1
```

3. 验证所创建的 VXLAN 链路。

```
# dladm show-vxlan
LINK  ADDR      VNI  MGROUP
vxlan1 10.10.11.1 10   224.0.0.1
```

因为您尚未指定多播地址，所以此 VXLAN 网段使用 All Host 多播地址来寻址同一个网段上的所有主机。

4. 检查 VXLAN 链路信息。

```
# dladm show-link vxlan1
LINK  CLASS MTU  STATE OVER
vxlan1 vxlan 1440 up   --
```

vxlan1 已创建并且链路状态为 up。

5. 在 vxlan1 上创建 VNIC。

```
# dladm create-vnic -l vxlan1 vnic1
```

6. 验证所创建的 VNIC。

```
# dladm show-vnic
LINK  OVER  SPEED  MACADDRESS      MACADDRTYPE  VIDS
vnic1 vxlan1 10000  2:8:20:fe:58:d4 random        0
```

7. 在 VNIC 上配置 IP 接口。

```
# ipadm create-ip vnic1

# ipadm create-addr -T static -a local=10.10.12.1/24 vnic1/v4

# ipadm show-addr vnic1
ADDROBJ  TYPE  STATE  ADDR
```

```
vnic1/v4 static ok    10.10.12.1/24
```

您已成功通过指定 IP 地址创建 VXLAN。您已在 VXLAN 上创建一个 VNIC 并且已配置 IP 接口。

例 3-2 将在 VXLAN 上创建的 VNIC 分配给区域并配置 IP 接口

此示例假定您已完成例 3-1 “创建 VXLAN 并为在 VXLAN 上创建的 VNIC 配置 IP 接口”中的步骤 1 到 6。

在创建 VNIC 后，将 VNIC 分配给区域并配置 IP 接口。

```
global# zonecfg -z zone2
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vnic1
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z zone2 reboot

global# zlogin zone2
zone2# ipadm create-ip vnic1
zone2# ipadm create-addr -a 192.168.3.85/24 vnic1
ipadm: vnic1/v4

zone2# exit
```

您已将 VNIC 分配给区域，并且随后在 VNIC 上配置了 IP 接口。

显示 VXLAN 信息

可以使用 `dladm show-link` 命令查看有关 VXLAN 链路的一般链路信息。要查看特定于 VXLAN 的信息，请使用 `dladm show-vxlan` 命令。

```
# dladm show-link
LINK          CLASS  MTU   STATE  OVER
net6          phys   1500  down   --
net0          phys   1500  up     --
net2          phys   1500  unknown --
net3          phys   1500  unknown --
net1          phys   1500  unknown --
net5          phys   1500  unknown --
net4          phys   1500  up     --
vxlan1       vxlan  1440  up     --
vnic1        vnic   1440  up     vxlan1

# dladm show-vxlan vxlan1
LINK          ADDR          VNI  MGROUP
```

```
vxlan1          10.10.11.1    10    224.0.0.1
```

删除 VXLAN

要删除 VXLAN 链路，请使用 `dladm delete-vxlan` 命令。在删除某个 VXLAN 链路之前，必须通过使用 `dladm show-link` 命令确保该 VXLAN 链路上未配置任何 VNIC。

以管理员身份发出以下命令：

```
# dladm delete-vxlan VXLAN
```

例如，如果要删除 `vxlan1`，则键入以下命令：

```
# dladm delete-vxlan vxlan1
```

将 VXLAN 分配给区域

可以通过将 VXLAN 作为底层链路分配给区域的 `anet` 资源来创建属于 VXLAN 网段的区域。有关配置区域的信息，请参见《[创建和使用 Oracle Solaris 区域](#)》。

▼ 如何将 VXLAN 分配给区域

1. 成为管理员。
有关更多信息，请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“使用所指定的管理权限”。

2. 确定系统上的可用 IP 地址。

```
# ipadm show-addr
```

3. 通过指定 IP 地址创建 VXLAN。

```
# dladm create-vxlan -p prop=value VXLAN-LINK
```

4. 验证所创建的 VXLAN。

```
# dladm show-vxlan
```

5. 通过将所创建的 VXLAN 分配为区域的 `anet` 的底层链路来配置区域。

```
global# zonecfg -z ZONE
zonecfg:zone2> add anet
zonecfg:zone2:net> set linkname=datalink
```

```

zonecfg:zone2:net> set lower-link=VXLAN-LINK
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z zone reboot

```

将 VXLAN 分配为区域的 anet 的底层链路。

例 3-3 将 VXLAN 分配给区域的 anet

```

# ipadm show-addr net4
ADDROBJ  TYPE  STATE  ADDR
net4/v4  static ok      10.10.11.1/24 2

# dladm create-vxlan -p addr=10.10.11.1,vni=10 vxlan1

# dladm show-vxlan
LINK  ADDR  VNI  MGROUP
vxlan1 10.10.11.1 10  224.0.0.1

```

因为您尚未指定多播地址，所以此 VXLAN 网段使用 All Host 多播地址来寻址同一个网段上的所有主机。

```

# dladm show-link vxlan1
LINK  CLASS MTU  STATE OVER
vxlan1 vxlan 1440 up  --

```

vxlan1 已创建并且链路状态为 up。

```

global# zonecfg -z zone2
zonecfg:zone2> add anet
zonecfg:zone2:net> set linkname=net1
zonecfg:zone2:net> set lower-link=vxlan1
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z zone2 reboot

```

将 vxlan1 分配为区域的 anet 的底层链路。

当区域启动时，会在 vxlan1 上的 zone2 中创建 net1。

使用案例：在链路聚合上配置 VXLAN

以下使用案例说明如何完成以下任务：

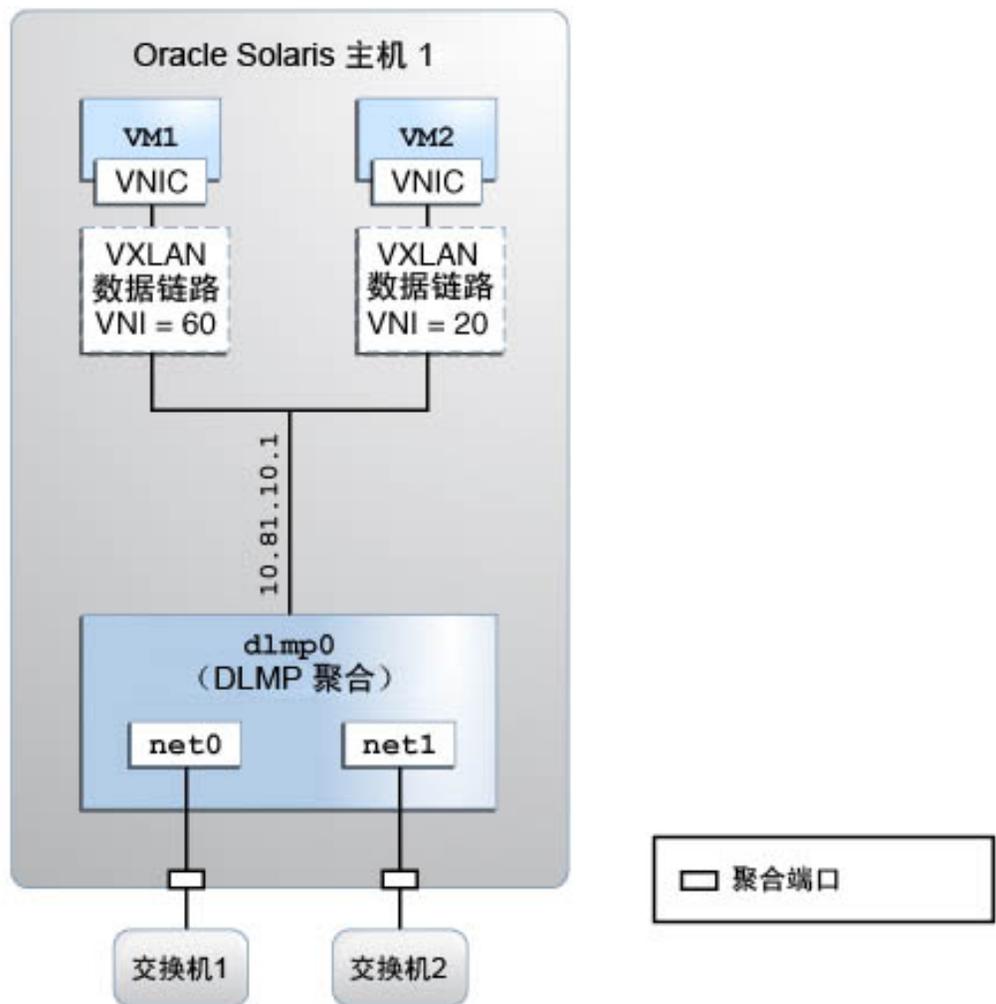
- 创建 DLMP 聚合
- 在聚合上配置 IP 地址

- 在聚合上创建两个 VXLAN
- 配置两个使用 VXLAN 数据链路作为下行链路的区域

有关链路聚合的信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的第 2 章“使用链路聚合配置高可用性”。

下图显示了 DLMP 聚合上的 VXLAN 配置。

图 3-3 链路聚合上的 VXLAN



当聚合端口或外部交换机出故障时，只要至少一个端口和交换机正常工作，该聚合上的 VXLAN 数据链路就会继续存在，从而在故障转移期间提供网络高可用性。例如，如果 net0 出现故障，则 DLMP 聚合将在 VXLAN 数据链路之间共享剩余端口 net1。聚合端口间的分发是对用户透明的，并独立于连接到聚合的外部交换机。

1. 成为管理员。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 显示数据链路信息以标识聚合的数据链路。

```
# dladm show-link
LINK      CLASS    MTU     STATE   OVER
net0     phys    1500    up      --
net1     phys    1500    up      --
net2     phys    1500    up      --
```

3. 确保您希望聚合的数据链路上未配置 IP 接口。如果在任何链路上配置了任何接口，则删除该接口。

```
# ipadm show-if
IFNAME    CLASS    STATE   ACTIVE  OVER
lo0       loopback ok      yes     --
net0      ip       ok      no      --
# ipadm delete-ip net0
```

4. 使用链路 net0 和 net1 创建 DLMP 聚合。

```
# dladm create-aggr -m dlmp -l net0 -l net1 dlmp0
```

5. 在聚合 dlmp0 之上配置 IP 接口。

```
# ipadm create-ip dlmp0
# ipadm create-addr -T static -a local=10.10.10.1 dlmp0/v4
```

6. 通过指定在该聚合上配置的 IP 地址来创建两个 VXLAN，另外，指定 VXLAN 网段的网络标识符 VNI。

```
# dladm create-vxlan -p addr=10.10.10.1,vni=20 vxlan20
# dladm create-vxlan -p addr=10.10.10.1,vni=60 vxlan60
```

两个 VNI 都配置了缺省多播地址。

7. 配置使用 VXLAN 数据链路 vxlan20 作为下行链路的区域 VM1。

```
global# zonecfg -z VM1
zonecfg:VM1> add anet
zonecfg:VM1:net> set linkname=net0
zonecfg:VM1:net> set lower-link=vxlan20
zonecfg:VM1:net> end
zonecfg:VM1> verify
zonecfg:VM1> commit
```

```
zonecfg:VM1> exit
global# zoneadm -z VM1 reboot
```

8. 配置使用 VXLAN 数据链路 vxlan60 作为下行链路的区域 VM2。

```
global# zonecfg -z VM2
zonecfg:VM2> add anet
zonecfg:VM2:net> set linkname=net0
zonecfg:VM2:net> set lower-link=vxlan60
zonecfg:VM2:net> end
zonecfg:VM2> verify
zonecfg:VM2> commit
zonecfg:VM2> exit
global# zoneadm -z VM2 reboot
```

将 net0 和 net1 数据链路聚合到 DLMP 聚合 dlm0，并为该聚合配置 IP 地址 10.10.10.1。在为该聚合配置的指定 IP 地址 10.10.10.1 上创建 VXLAN vxlan20 和 vxlan60。在 VXLAN 网段 20 上创建 VXLAN vxlan20，并在 VXLAN 网段 60 上创建 VXLAN vxlan60。为区域 VM1 配置 VXLAN 数据链路 vxlan20 作为下行链路，并为区域 VM2 配置 VXLAN 数据链路 vxlan60 作为下行链路。

使用边缘虚拟桥接管理服务器-网络边缘虚拟化

服务器-网络边缘存在于服务器端口与它的第一中继站交换机端口之间的连接中。在此边缘上的服务器端口和交换机端口上，诸如虚拟局域网 (virtual local area network, VLAN) 和链路聚合控制协议 (Link Aggregation Control Protocol, LACP) 之类的网络配置必须相同。您可以使用数据中心桥接交换 (Data Center Bridging Exchange, DCBX) 功能在服务器和交换机端口上自动进行配置。有关更多信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的第 6 章“使用数据中心桥接管理聚合网络”。

通过服务器虚拟化，可以将多个虚拟端口关联到服务器端口后的虚拟机 (virtual machine, VM)，而不仅仅是连接到交换机端口的一个服务器端口。服务器虚拟化对服务器-网络边缘还存在以下额外的要求：

- 通过外部交换机在虚拟机之间进行切换，从而使 VM 间的通信遵守交换机上配置的策略
- 将虚拟端口属性扩展到网络中

Oracle Solaris 支持边缘虚拟桥接 (edge virtual bridging, EVB)，这是一个发展中的 IEEE 标准，旨在解决这些需求。

本章包含以下主题：

- “服务器-网络边缘虚拟化中的 EVB 支持” [67]
- “通过使用 EVB 改进网络和服务器效率” [68]
- “安装 EVB” [71]
- “控制基于同一物理端口的 VM 之间的切换” [71]
- “使用 VDP 交换 VNIC 信息” [77]
- “显示 VDP 和 ECP 状态和统计数据” [79]
- “更改缺省 EVB 配置” [80]

服务器-网络边缘虚拟化中的 EVB 支持

一个虚拟化服务器在同一物理链路上可以包含多个虚拟 NIC。可以将这些 VNIC 指定给 VM。通常，交换机不会在接收包的同一链路上传回包。VM 之间的包通过主机内部的虚拟交换机环回。因此，在外部交换机上配置的任何策略都不适用于 VM 之间的包。借

助 EVB 支持，在对 VM 之间的包强制执行任何策略后，Oracle Solaris 和交换机将允许 VM 之间的包通过外部交换机进行交换。有关 VNIC 的更多信息，请参见[第 2 章 创建和管理虚拟网络](#)。

此外，Oracle Solaris 借助 EVB 支持，还可以和交换机交换有关 VNIC 的信息。通过这种信息交换，交换机就可以自动配置 VNIC 属性，如网络中的带宽限制、带宽共享和 MTU。如果没有这种功能，每次在服务器上创建、修改或删除 VNIC 时，服务器管理员和网络管理员就必须相互协调，以便对交换机做出相应更改。将 VNIC 属性扩展到网络后，就可以根据 VNIC 属性有效地使用网络资源。例如，在包到达主机后对包执行带宽限制就不是很有用，因为这些包可能已占用了链路带宽。

反射中继

反射中继功能可让使用同一物理 NIC 上的 VNIC 的 VM 通过外部交换机进行通信。交换机必须支持此功能。在 Oracle Solaris 中，LLDP 扩展为包括 EVB 类型长度值 (type-length value, TLV) 单元，该单元用于确定交换机是否支持反射中继功能，以及在交换机上启用还是禁用反射中继功能。因此，只有在交换机支持 LLDP 和 EVB TLV 单元的情况下，才能使用 LLDP 在交换机上自动检测和配置此功能。否则，必须在交换机上手动配置反射中继功能。有关如何手动配置反射中继的信息，请参阅交换机制造商的文档。

有关 Oracle Solaris 中反射中继支持的更多信息，请参见[“控制基于同一物理端口的 VM 之间的切换” \[71\]](#)。有关 LLDP TLV 单元的更多信息，请参见[《在 Oracle Solaris 11.2 中管理网络数据链路》中的“LLDP 代理通告的信息”](#)。

在网络中自动进行 VNIC 配置

Oracle Solaris 使用在 IEEE 802.1Qbg 中定义的虚拟站接口搜索和配置协议 (VSI discovery and configuration protocol, VDP) 与交换机交换 VNIC 信息。如果交换机支持 VDP，则会在交换机上自动配置 VNIC 属性。这对于使用 DCBX 交换物理链路属性的主机与交换机是类似的。在创建、修改或删除 VNIC 时，将启动主机和交换机之间的 VDP 交换。这种交换可使交换机根据 VNIC 的属性分配以 VNIC 为目标的包的资源。

有关在 Oracle Solaris 中在系统与外部交换机之间交换 VNIC 信息的更多信息，请参见[“使用 VDP 交换 VNIC 信息” \[77\]](#)和[“VDP 如何交换 VNIC 信息” \[78\]](#)。

通过使用 EVB 改进网络和服务器效率

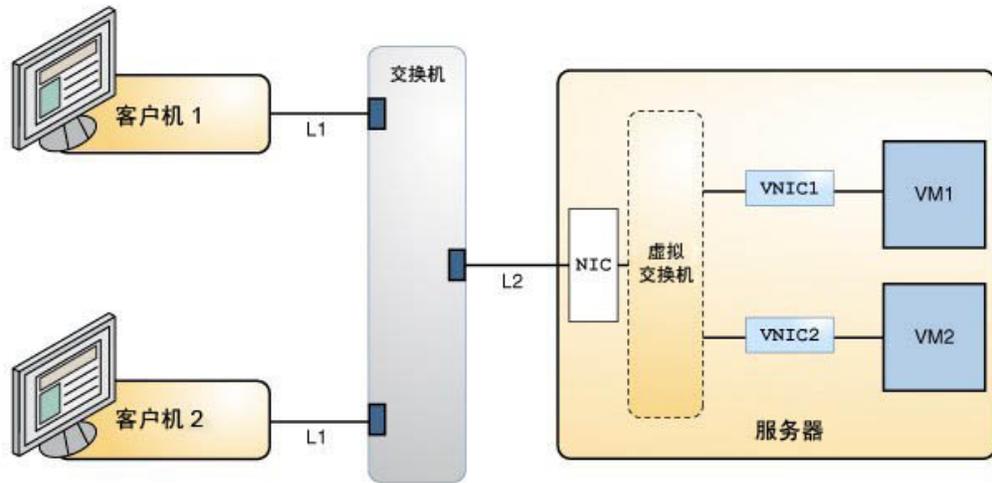
本节提供了一个示例来说明当在服务器上启用 EVB 时如何会提高服务器和网络效率。

此示例假定服务器在云环境中在同一物理计算机上托管着两个应用程序。

- 各个应用程序作为物理计算机上的单独虚拟机 (VM1 和 VM2) 托管于云上。分别为 VM1 和 VM2 配置了 VNIC VNIC1 和 VNIC2。
- 具有帐户的客户机 (客户机 1 和客户机 2) 可以访问应用程序。
- 虚拟机 (VM1 和 VM2) 共享物理系统的资源和链路 L2 上的带宽。
- 客户机使用链路 L1 连接到交换机。交换机使用链路 L2 连接到 NIC。
- 预先确定的 SLA 决定了为虚拟机指定的资源。对于虚拟机的 SLA, 包括以下 (L2) 带宽使用：
 - VM1 在运行高优先级的传输控制协议 (Transmission Control Protocol, TCP) 服务。因此, VM1 的 SLA 的最大带宽限制为 8 Gbps。
 - VM2 在运行优先级不高的用户数据报协议 (User Datagram Protocol, UDP) 服务。因此, VM2 的 SLA 的最大带宽限制为 3 Gbps。

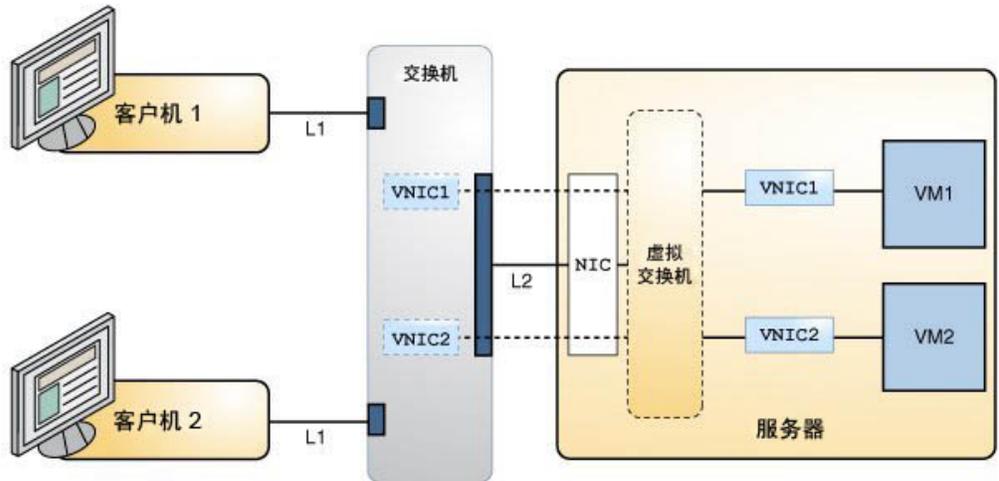
下图显示了服务器上托管的应用程序。

图 4-1 未使用 EVB 的应用程序设置



当在服务器和交换机上启用了 EVB 时, 服务器通过同一个物理交换机端口与交换机交换 VNIC 信息, 如下图所示。

图 4-2 启用了 EVB 的应用程序设置



下表显示了在服务器和交换机上启用 EVB 之前和之后的服务器效率。

表 4-1 未使用 EVB 时和使用 EVB 时的服务器效率

未使用 EVB 时的服务器效率	使用 EVB 时的服务器效率
服务器对来自客户机的传入通信进行控制以实现带宽控制。	交换机对以服务器为目的地的通信进行控制。
使用了系统资源，所以会影响系统和网络性能。	不使用系统资源来处理带宽，因而提高了系统效率。
<p>在此示例中，当客户机（客户机 1 和客户机 2）需要同时使用服务器时，客户机将使用链路 L2 的带宽和服务器资源。服务器在 VM1 和 VM2 的 VNIC 上强制实现 SLA 以控制客户机的传入和传出通信。但是，网络性能和带宽使用在以下方面受影响：</p> <ul style="list-style-type: none"> 来自客户机（客户机 1 和客户机 2）的通信无限制地使用链路 L2 的带宽。此外，如果主机上配置了带宽限制，则可能会在该主机上丢弃使用 L2 带宽的包，这将导致无法高效使用该带宽。 VM1 提供高优先级的 TCP 服务，而 VM2 提供优先级不高的 UDP 服务。在服务器上控制 VM1 的带宽将导致 TCP 进行响应，从而影响 VM1 对链路 L2 上带宽的使用。不过，控制 VM2 在服务器上的服务不会影响其对链路 L2 带宽的使用。这会影响到使用链路 L2 的其他服务。 	<p>在服务器和交换机上启用了 EVB 时，会在以下方面提高系统效率：</p> <ul style="list-style-type: none"> 在服务器的 VNIC 上配置的 SLA 将反映在交换机上。 交换机根据所配置的带宽对以 VM1 和 VM2 为目的地的通信进行控制，因此帮助恰当地利用链路 L2 的带宽，从而提高了网络效率。 <p>由于交换机对带宽进行控制，服务器无需处理接收端的带宽，因此提高了服务器效率。</p>
在此示例中，传入到服务器的 UDP 和 TCP 服务的网络通信无限制地使用链路 L2 上的可用带宽。服务器在收到网络通信后，将根据所配置的带宽限制来控制网络通信。	除服务器之外，交换机也对所配置的带宽限制（3 Gbps 和 8 Gbps）进行控制。因此，将根据所配置的带宽限制使用共享链路 L2。

安装 EVB

必须安装 EVB 软件包才能在系统上使用 EVB。

▼ 如何安装 EVB

1. 成为管理员。
有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 验证是否已安装了 EVB 软件包。

```
# pkg info evb
```

3. 如果未安装 EVB 软件包，请安装此软件包。

```
# pkg install evb
```

4. 验证是否已启用了此服务。

```
# svcsvd vdp
```

5. 如果未启用此服务，请启用此服务。

```
# svcadm enable vdp
```

安装 EVB 软件包后，会自动启用缺省 EVB 配置。通过接受缺省 EVB 配置，系统可以立即与外部交换机交换有关您在系统上配置的任何 VNIC 的信息。

- 另请参见
- 要了解有关 VNIC 信息交换、用于交换 VNIC 信息的协议以及 EVB 组件的更多信息，请参见“使用 VDP 交换 VNIC 信息” [77]。
 - 要显示在系统上启用了 EVB 时物理以太网链路的 VSI 发现和配置协议 (VSI discovery and configuration protocol, VDP) 状态的相关信息，并检查是否正在为 VNIC 交换 VDP 包，请参见“显示 VDP 和 ECP 状态和统计数据” [79]。
 - 要更改缺省 EVB 配置，请参见“更改缺省 EVB 配置” [80]。
 - 要显示缺省 EVB 配置信息，请参见例 4-2 “显示物理链路上与 EVB 相关的数据链路属性”。

控制基于同一物理端口的 VM 之间的切换

可以使用 `vswitchmode` 数据链路属性控制基于同一物理端口的 VM 之间的切换。三个可能的值是：

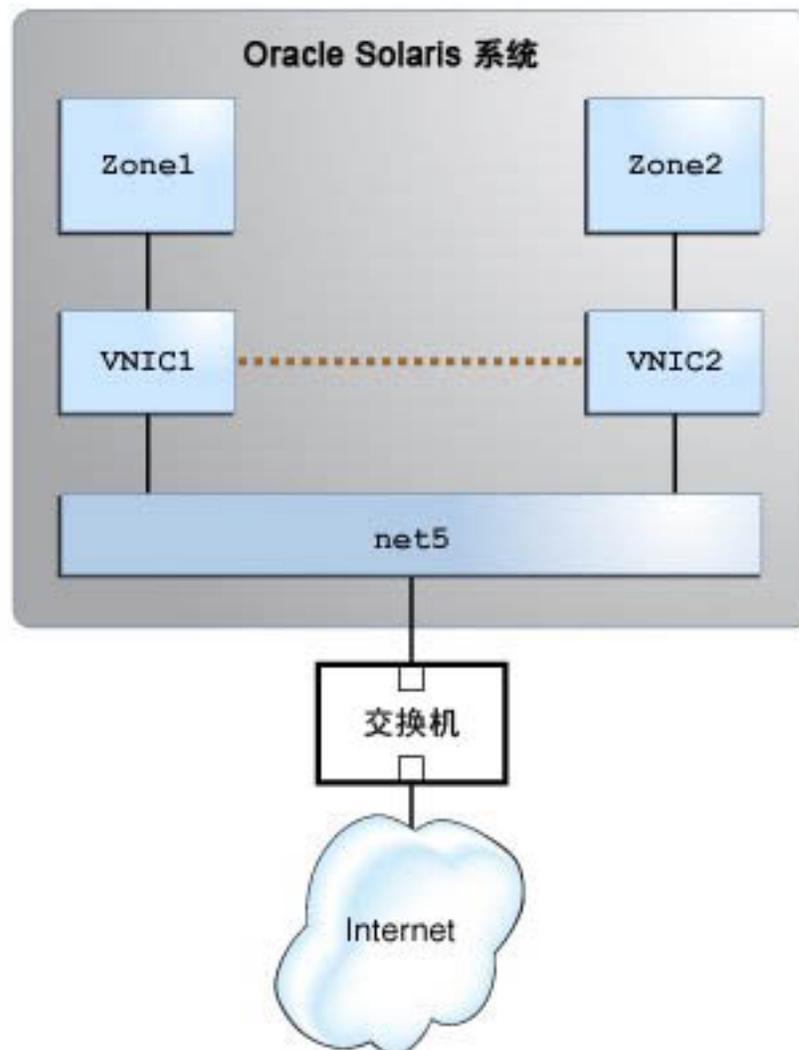
- local – 使基于同一物理 NIC 的 VM 之间的网络通信可以在内部交换。这是缺省模式。
- remote – 使基于同一物理 NIC 的 VM 之间的网络通信可以通过外部交换机进行交换。
- auto – 使用 LLDP 确定外部交换机是否支持反射中继。如果外部交换机支持反射中继，则通过外部交换机交换 VM 之间的网络通信。否则，在内部交换 VM 之间的网络通信。

允许 VM 通过外部交换机通信

如果基于同一个物理 NIC 配置了多个 VNIC，则可以将 vswitchmode 数据链路属性设置为 remote，以便在外部通过交换机发送网络通信。不过，必须将外部交换机配置为反射中继模式。启用了反射中继的交换机配置特定于交换机类型。有关更多信息，请参阅交换机制造商的文档。

下图显示了一个样例系统，该系统具有连接到外部交换机的 10G 以太网链路并且托管着为同一客户运行服务的两个区域 (VM)。

图 4-3 区域之间的内部通信



因为 Zone1 和 Zone2 这两个区域为同一客户运行服务，所以这两个区域可以在内部通信，而无任何限制。因此，可以在内部交换 VNIC1 与 VNIC2 之间的通信。

您将检查物理 NIC net5 的 vswitchmode 属性的现有值，如下所示：

```
# dladm show-linkprop -p vswitchmode net5
LINK PROPERTY PERM VALUE EFFECTIVE DEFAULT POSSIBLE
net4 vswitchmode rw local local local local,remote,auto
```

对于 VALUE 和 EFFECTIVE 字段，该输出显示了 local。此值表示区域之间的通信是内部的。

在此示例中，假设 Zone1 和 Zone2 两个区域需要为不同客户运行服务，并且外部交换机配置了访问控制列表 (access control list, ACL)，该列表控制这些服务的网络通信。因此，它们不能在内部进行通信，VNIC1 与 VNIC2 之间的网络通信必须通过交换机在外部进行交换。

因而，您必须通过将 vswitchmode 属性设置为 remote 来禁用区域之间的内部通信，如下所示：

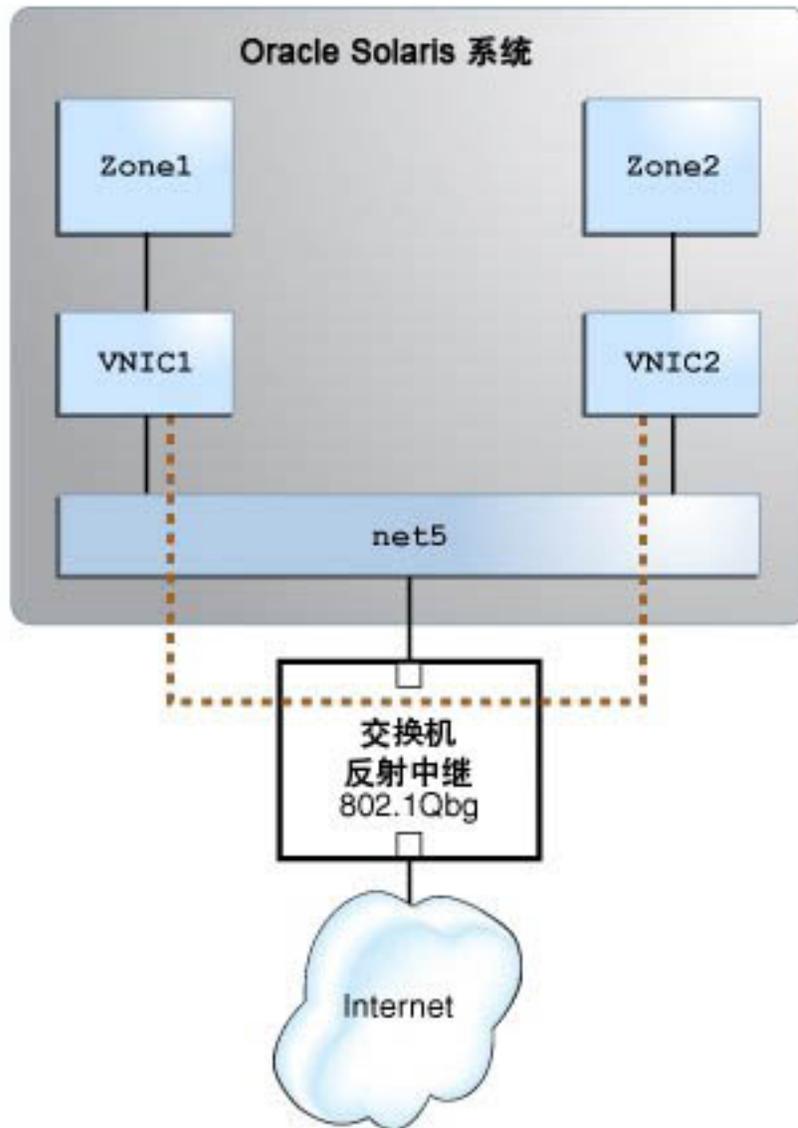
```
# dladm set-linkprop -p vswitchmode=remote net5

# dladm show-linkprop -p vswitchmode net5
LINK PROPERTY PERM VALUE EFFECTIVE DEFAULT POSSIBLE
net5 vswitchmode rw remote remote local local,remote,auto
```

注 - 在将 vswitchmode 设置为 remote 之前，必须为外部交换机配置反射中继。

因为您将 vswitchmode 属性设置为 remote 来禁用 VNIC 的内部通信，所以将通过外部交换机发送 VNIC 之间的网络通信，如下图所示。

图 4-4 使用外部交换机的区域间通信



使用 LLDP 管理 VM 之间的通信

您可以使用 LLDP 自动配置 VM 之间的通信。LLDP 根据外部交换机是否支持反射中继，将网络通信的交换配置为内部或外部。要使用 LLDP，请将 `vswitchmode` 数据链路属性设置为 `auto`。首先，您必须确保以下各项：

- LLDP 软件包已安装。
要检查 LLDP 软件包是否已安装，请使用下面的命令：

```
# pkg info lldp
```

- LLDP 服务处于联机状态。
要检查 LLDP 服务是否处于联机状态，请使用下面的命令：

```
# svcs lldp
STATE      STIME     FMRI
online      Jul_13    svc:/network/lldp:default
```

- 已在 `dot1-tlv` TLV 单元中启用 EVB。
- NIC 的 LLDP 模式为 `both`。
在此示例中，要检查是否在 `dot1-tlv` TLV 单元中启用了 EVB 以及 LLDP 模式是否为 `both`，可以使用下面的命令：

```
# lladm show-agentprop -p mode,dot1-tlv net5
AGENT  PROPERTY  PERM  VALUE  DEFAULT  POSSIBLE
net5   mode      rw    both   disable  txonly,rxonly,both,disable
net5   dot1-tlv  rw    evb    none     none,vlanname,pvid,linkaggr,pfc,
                                     appln,evb,etscfg,etsreco,all
```

将 `vswitchmode` 数据链路属性设置为 `auto`：

```
# dladm set-linkprop -p vswitchmode=auto net5
```

将 `vswitchmode` 数据链路属性设置为 `auto` 时，可以使用 `dladm show-linkprop` 命令的输出检查 VM 之间的通信是内部的还是通过外部交换机的。

```
# dladm show-linkprop -p vswitchmode net5
LINK  PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net5  vswitchmode  rw    auto   remote    local    local,remote,auto
```

由于此输出的 `EFFECTIVE` 字段的值为 `remote`，因此 LLDP 已在外部交换机上启用反射中继，VM 之间的通信是通过外部交换机进行的。

有关 LLDP 的更多信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的第 5 章“使用链路层发现协议交换网络连接信息”。

使用 VDP 交换 VNIC 信息

系统（站）与外部交换机（网桥）之间的 VNIC (VSI) 信息交换是使用 VSI 搜索和配置协议 (VSI discovery and configuration protocol, VDP) 进行的。VDP 类型长度值 (type-length value, TLV) 单元是使用边缘控制协议 (Edge Control Protocol, ECP) 进行交换的，该协议能够可靠地在对等点之间传输 VDP 包。创建或删除 VNIC 时会交换 VDP TLV 单元。

以下 EVB 组件使系统能够向外部交换机通告 VNIC (VSI) 信息：

- VSI 配置文件包含为特定 VNIC 配置的链路属性。因此，系统最多可以拥有与已配置的 VNIC 同等数量的 VSI 配置文件。
- VSI 标识符唯一标识 VSI 实例。在 Oracle Solaris 中，此 VSI 实例是 VNIC (VSI) 的 MAC 地址。VSI 类型 ID 和 VSI 版本标识给定 VSI 管理器 ID 内的配置文件。
- VSI 管理器通过使用一组特定的 VNIC 属性来映射 VSI 类型 ID 与 VSI 版本，从而管理系统上的多个 VSI 配置文件。Oracle Solaris 已将缺省 VSI 管理器 `oracle_v1` 定义为 3 字节编码。此 3 字节编码在 VDP 包中被 Oracle Solaris 主机用作 VSI 类型 ID。
- VSI 管理器 ID 标识与特定的 VSI 类型 ID/VSI 版本对相关的 VSI 管理器。VSI 管理器 ID 表示为 IPv6 地址。Oracle Solaris 已定义了缺省 VSI 管理器 ID `ORACLE_VSIMGR_V1`。

注 - 目前，没有已定义的标准用于定义 VSI 配置文件及其具体属性。VSI 类型的定义特定于供应商并且与 VSI 管理器 ID 紧密关联。

此 `oracle_v1` 编码支持以下属性：

- 带宽限制
- 带宽份额
- 底层链路的链路速度
- VNIC 的最大传输单元 (maximum transmission unit, MTU)

在 Oracle Solaris 中，系统使用 `oracle_v1` 编码来对链路信息进行编码，然后将此信息传输到外部交换机。交换机在收到此信息后，将使用相同的 `oracle_v1` 编码来解码已编码的信息。

缺省情况下，Oracle Solaris 主机会将以下元素发送到外部交换机：

- Oracle VSI 管理器 - `oracle_v1`
- VSI 类型 ID - 使用 `oracle_v1` 编码进行编码的 VNIC 属性
- VSI 版本 - 始终为 0

在 Oracle Solaris 中，VNIC 信息交换机制如下所述：

1. 外部交换机配置为支持 Oracle VSI 管理器 `oracle_v1`。

2. 外部交换机使用 `oracle_v1` 确定使用 VSI 类型 ID 编码的属性。
3. 外部交换机将属性配置应用于该 VNIC 的包。

特定于 Oracle 组织的 OUI TLV 单元遵循 VSI 管理器 ID TLV，以表明它是特定于 Oracle 的 VSI 管理器 ID。来自交换机的响应中缺少特定于 Oracle 的 TLV 单元将向 Oracle Solaris 主机表明交换机不支持 Oracle VSI 管理器（编码）。Oracle Switch ES1-24 支持 Oracle VSI 管理器 `oracle_v1`。有关 Oracle Switch ES1-24 上的 EVB 配置的更多信息，请参见《[Sun Ethernet Fabric Operating System, EVB Administration Guide](#)》（《Sun 以太网光纤操作系统 EVB 管理指南》）。

注 - 除了支持 VDP 和 ECP 协议之外，要与 Oracle Solaris 系统交互操作，外部交换机还必须支持 `ORACLE_VSIMGR_V1`（缺省 Oracle VSI 管理器 ID）和 Oracle 组织唯一标识符（organizationally unique identifier, OUI）TLV（子类型 `VDP_ORACLEOUI_VSIMGR_SUBTYPE`，用于携带编码信息）。

VDP 如何交换 VNIC 信息

VNIC 信息交换的工作方式如下所述：

系统通过指定 VNIC 及其关联的配置文件向外部交换机发送一个关联请求 (ASSOC)。外部交换机使用成功或失败响应对关联请求进行响应。系统随后可以向外部交换机发送取消关联请求 (DEASSOC) 以删除 VNIC 的关联。有关如何显示和获取 VNIC 的请求状态的信息，请参见“[显示 VDP 和 ECP 状态和统计数据](#)” [79]。

创建 VNIC 时，VDP 交换按下述方式进行：

1. 系统向外部交换机发送一个包含 VNIC 相关信息的 VDP 关联 (ASSOC) 请求 TLV 单元。
2. 外部交换机接收 VDP (ASSOC) TLV 单元，并通过使用 VSI 类型 ID、VSI 版本和 VSI 管理器 ID 获取 VNIC 信息。
3. 外部交换机为 VNIC 应用属性配置。
4. 外部交换机向系统发送一个 VDP 关联 (ASSOC) 响应 TLV 单元，指明外部交换机已为 VNIC 配置了属性。

删除 VNIC 时，将按下述方式进行 VDP 交换：

1. 系统向外部交换机发送一个包含 VSI ID 的 VDP 取消关联 (DEASSOC) 请求 TLV 单元。
2. 外部交换机接收 VDP (DEASSOC) TLV 单元，并获取删除的 VSI 的 VSI ID。
3. 外部交换机删除已删除 VNIC 的配置。
4. 外部交换机向系统发送 VDP 取消关联 (DEASSOC) 响应 TLV 单元。

注 - 在 Oracle Solaris 中，VDP 仅支持 ASSOC 和 DEASSOC VDP 请求。

显示 VDP 和 ECP 状态和统计数据

您可以显示物理以太网链路的 VDP 状态的相关信息，例如系统上是否已启用 EVB 以及是否正在为 VNIC 交换 VDP 包。要仅显示单个链路的信息，请在命令中指定该链路。否则，将显示所有以太网链路的 VDP 信息。

显示 VDP 状态和统计数据

要显示 VDP 状态，请键入以下命令：

```
# dladm show-ether -P vdp
VSI    LINK    VSIID          VSI-TYPEID    VSI-STATE    CMD-PENDING
vnic1  net0    2:8:20:22:3c:6b  98/0          ASSOC        NONE
vnic2  net0    2:8:20:90:7f:ef  96/0          ASSOC        NONE
```

VSI-STATE 显示与对等点的 VDP 交换的状态。可能的值是：

- TIMEDOUT – 对等点尚未响应 VDP 请求。
- ASSOC – 对等点已成功处理请求。
- DEASSOC – 主机或对等点已拒绝请求。如果对等点无法确定指定的配置文件或属性，则对等点可以拒绝请求。如果主机使用 oracle_v1 编码但对等点未在其响应中包括 Oracle OUI，则主机可以拒绝交换 VDP 包。

示例输出显示，在链路 net0 上配置了两个 VSI (VNIC)。其特定的 VSI ID 指向各自的 MAC 地址。VNIC vnic1 和 vnic2 的 VSI-TYPE ID 是基于其各自的属性（带宽限制和 MTU）生成的，并且编码是由 oracle_v1 定义的。

要获得有关传出或传入的 VDP 包的统计数据，请键入以下命令：

```
# dlstat show-ether -P vdp net1
LINK    IPKTS    OPKTS    KeepAlives
net1    3         2         1
```

显示链路属性

可以使用 `dladm show-linkprop` 命令的 `-p` 选项显示链路属性。

以下示例说明了如何显示 vnic1 和 vnic2 的链路属性。

```
# dladm show-linkprop -p maxbw,mtu vnic1
```

```

LINK      PROPERTY      PERM  VALUE    EFFECTIVE  DEFAULT  POSSIBLE
vnic1    maxbw           rw    100     100       --       --
vnic1    mtu             rw    1500    1500     1500    1500

# dladm show-linkprop -p maxbw,mtu vnic2
LINK      PROPERTY      PERM  VALUE    EFFECTIVE  DEFAULT  POSSIBLE
vnic2    maxbw           rw    20      20        --       --
vnic2    mtu             rw    1500    1500     1500    1500

```

显示 ECP 状态和统计数据

VDP 使用 ECP 来交换消息。以下示例说明了特定于物理链路 net0 的 ECP 的状态。

```

# dladm show-ether -P ecp net0
LINK      MAX-RETRIES  TIMEOUT
net0      3            164

```

MAX-RETRIES 指定 ECP 在未获得来自对等点的确认时传输包的次数。

TIMEOUT 指定重新传输包之前的时间间隔（毫秒）。ECP 在重新传输包之前等待确认的时间间隔。

要获取物理链路的统计数据，请键入以下命令：

```

# dlstat show-ether -P ecp
LINK      IPKTS  OPKTS  IERRORS  OERRORS  RETRANSMITS  TIMEOUTS
net0      3      2      0        0        1            0

```

更改缺省 EVB 配置

缺省情况下，您不需要更改缺省 EVB 配置。在多数情况下，您都可以安装 EVB，并使用缺省 EVB 配置来与外部交换机交换有关您在系统上配置的任何 VNIC 的信息。不过，如果您希望完全控制和管理主机和网络上的 EVB 配置，则可以更改缺省配置。

当使用缺省 Oracle Solaris VSI 管理器 ID ORACLE_VSIMGR_V1 时，系统将自动为您创建的 VNIC 生成 VSI 类型 ID。因此，无需设置数据链路属性，例如 vsi-typeid 和 vsi-vers。但是，如果您未使用缺省 VSI 管理器 ID，则必须通过使用 dladm set-linkprop 命令设置与 EVB 相关的数据链路属性。要设置与 EVB 相关的数据链路属性，外部交换机必须能够与系统通信并检索一组给定 VSI 类型 ID 和 VSI 版本的属性。

在使用 EVB 时请使用缺省 Oracle VSI 管理器 ID，这样 Oracle VSI 管理器可自动为系统的 VSI 配置文件生成 VSI 类型 ID 和 VSI 版本。

您可以配置与 EVB 相关的以下数据链路属性：

- `vsi-mgrid` – 指定为物理链路或 VNIC 设置的 VSI 管理器 ID。如果没有为 VNIC 设置此属性，则将使用底层物理链路的缺省值 `ORACLE_VSIMGR_V1`。
如果显式设置了 `vsi-mgrid` 属性，则您还需要显式设置 VSI 类型 ID 和 VSI 版本。此外，您还需要显式为数据链路配置这些属性。

注 - 在 Oracle Solaris 中，当您手动配置 VSI 管理器 ID、VSI 类型 ID 和 VSI 版本时，不会自动配置相应的 VNIC 属性。

- `vsi-mgrid-enc` – 指示与 VSI 管理器 ID 关联的编码。缺省情况下，此属性设置为 `oracle_v1`。如果不希望将 `oracle_v1` 与 VSI 管理器 ID 关联，请将此属性的值设置为 `none`。当设置值 `none` 时，还要确保手动配置 VSI 管理器 ID、VSI 类型 ID 和 VSI 版本，因为它们不会自动生成。
- `vsi-typeid` – 指定 VSI 类型 ID。VSI 类型 ID 和要与 VSI 配置文件关联的 VSI 版本成对。如果 `vsi-mgrid` 和 `vsi-mgrid-enc` 使用缺省值，则将自动生成该 3 字节值。否则，您必须明确地为该属性指定一个值。
- `vsi-vers` – 指定 VSI 版本。VSI 版本和要与 VSI 配置文件关联的 VSI 类型 ID 成对。如果 `vsi-mgrid` 和 `vsi-mgrid-enc` 使用缺省值，则将自动生成该 1 字节值。否则，您必须明确地为该属性指定一个值。

可以使用 `dladm show-linkprop` 命令显示与 EVB 相关的属性。可以从 VNIC 相关链路属性各自的 `EFFECTIVE` 字段值获取这些属性的生效值。有关更多信息，请参见例 4-2 “显示物理链路上与 EVB 相关的数据链路属性”。

有关 EVB 组件的更多信息，请参见“使用 VDP 交换 VNIC 信息” [77]。有关 EVB 的更多信息，请参见 `evb(7P)` 手册页。

▼ 如何更改缺省 EVB 配置

必须只为物理链路配置 `vsi-mgrid` 和 `vsi-mgrid-enc` 属性。必须为 VNIC 配置其他与 EVB 相关的属性，例如 `vsi-typeid` 和 `vsi-vers`。

1. 成为管理员。
有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。
2. 使用配置文件数据库中提到的数据链路属性创建 VNIC。

```
# dladm create-vnic -l datalink -p maxbw=maxbw-value,priority=priority-value VNIC
```
3. 在物理链路上，将与 VSI 管理器 ID 关联的编码设置为 `none`，因为您未使用缺省 Oracle VSI 管理器 ID。

```
# dladm set-linkprop -p vsi-mgrid-enc=none datalink
```

4. 为物理链路上的 VSI 管理器 ID 设置 IPv6 地址。

```
# dladm set-linkprop -p vsi-mgrid=IPv6-address datalink
```

5. 为您已创建的 VNIC 设置 VSI 类型 ID 和 VSI 版本。

```
# dladm set-linkprop -p vsi-typeid=VSI-Type-ID,vsi-vers=VSI-Version VNIC
```

6. 验证为 VNIC 设置的属性。

```
# dladm show-linkprop VNIC
```

例 4-1 设置与 EVB 相关的数据链路属性

以下示例说明了如何设置与 EVB 相关的数据链路属性。此示例使用的系统具有您可使用 IPv6 地址 IP1 访问的配置文件。

假设 VSI 管理器 ID (IP1) 定义了以下配置文件：

- VSI 类型 ID : 2
- VSI 版本 : 1
- 数据链路属性 : maxbw=20、priority=5

1. 使用配置文件中提到的数据链路属性创建 VNIC。

```
# dladm create-vnic -l net0 -p maxbw=20,priority=5 vnic1
```

2. 在物理链路 net0 上，将与 VSI 管理器 ID 关联的编码设置为 none，因为您未使用缺省 Oracle VSI 管理器 ID。

```
# dladm set-linkprop -p vsi-mgrid-enc=none net0
```

3. 为物理链路 net0 上的 VSI 管理器 ID 设置 IPv6 地址 IP1。

```
# dladm set-linkprop -p vsi-mgrid=IP1 net0
```

4. 为 vnic1 设置 VSI 类型 ID 和 VSI 版本。

```
# dladm set-linkprop -p vsi-typeid=2,vsi-vers=1 vnic1
```

5. 验证为 vnic1 设置的属性。

```
# dladm show-linkprop vnic1
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
...						
vnic1	vsi-typeid	rw	2	2	--	--
vnic1	vsi-vers	rw	1	1	--	--
vnic1	vsi-mgrid	rw	IP1	IP1	--	--
vnic1	vsi-mgrid-enc	rw	--	none	oracle_v1	none,oracle_v1

...

vnic1 的 VDP ASSOC TLV 单元包含以下信息：

- VSI 管理器 ID = IP1
- VSI 类型 ID = 2
- VSI 版本 = 1

例 4-2 显示物理链路上与 EVB 相关的数据链路属性

以下示例显示了物理链路上与 EVB 相关的属性。

```
# dladm show-linkprop -p vsi-mgrid,vsi-mgrid-enc net4
LINK      PROPERTY          PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
net4      vsi-mgrid         rw  --          --            ::           --
net4      vsi-mgrid-enc     rw  --          --            oracle_v1    none,oracle_v1
```

该输出显示了 Oracle Solaris 中的缺省 EVB 配置。通过使用 oracle_v1 编码，将基于 VNIC 上配置的属性自动生成 VSI 类型 ID 和 VSI 版本。

例 4-3 显示 VNIC 上与 EVB 相关的属性

以下示例显示了 VNIC 上与 EVB 相关的属性。

```
# dladm show-linkprop vnic0
LINK      PROPERTY          PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
...
vnic0     vsi-typeid        rw  --          94            --           --
vnic0     vsi-vers          rw  --          0             --           --
vnic0     vsi-mgrid         rw  --          ::            --           --
vnic0     vsi-mgrid-enc     rw  --          oracle_v1     oracle_v1    none,oracle_v1
...
```

该输出将 vnic0 的有效编码显示为 oracle_v1。而 vsi-typeid 的 EFFECTIVE 值 94 则是自动生成的，并且对 vnic0 有效。

关于弹性虚拟交换机

从 Oracle Solaris 11.2 发行版开始，您可以使用 Oracle Solaris 弹性虚拟交换机 (Elastic Virtual Switch, EVS) 功能来管理跨越若干个物理计算机的多个虚拟交换机。本章概述了 Oracle Solaris 中的弹性虚拟交换机功能并包括以下主题：

- “弹性虚拟交换机 (Elastic Virtual Switch, EVS) 功能概述” [85]
- “EVS 组件” [90]
- “EVS 管理命令” [94]
- “使用 EVS 时必需的软件包” [98]
- “EVS 如何使用区域” [99]
- “使用 EVS 时的安全要求” [99]

弹性虚拟交换机 (Elastic Virtual Switch, EVS) 功能概述

当今的数据中心包括多台物理服务器，这些服务器托管着若干个通过网络结构连接的虚拟机 (virtual machine, VM)。在数据中心置备 VM 网络对于管理员来说是一项挑战，因为这涉及在 VM 之间设置虚拟网络，管理 MAC 地址和 IP 地址，以及管理 VLAN 和 VXLAN。除了确保 VM 的内部和外部网络连接以外，面临的其他挑战还有对 VM 及 VM 中的应用程序置备和实施服务级别协议 (service-level agreement, SLA)。这些 SLA 包括带宽限制和优先级。数据中心管理员还需要在共享公用网络基础结构的多个租户之间提供隔离。

为满足这些要求，Oracle Solaris 网络虚拟化功能使管理员能够跨数据中心管理虚拟交换机。虚拟交换机是作为第一类操作系统抽象概念公开的。这些虚拟交换机也称为弹性虚拟交换机，它们跨越多个物理服务器，并使系统管理员能够将它们作为单个虚拟交换机进行管理。

Oracle Solaris 中的虚拟交换机

虚拟交换机是便于虚拟机之间通信的实体。在 Oracle Solaris 中，当您在数据链路（例如，链路聚合、物理 NIC 或 etherstub）上创建 VNIC 时，会自动或隐式创建虚拟交换机。虚拟交换机使 VM 之间的通信（VM 间通信）在物理计算机内循环，并且不通过电

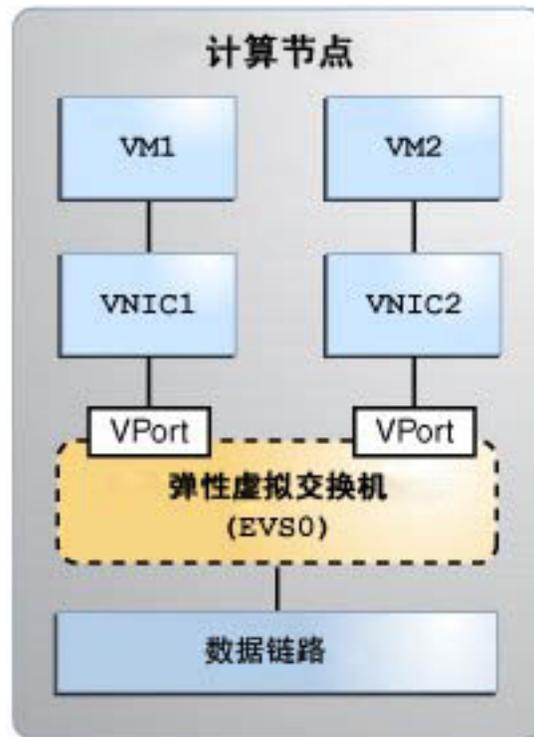
缆发出此通信。所有 VM 需要存在于同一第 2 层网段上才能相互通信。有关更多信息，请参见“[虚拟交换机](#)” [14]。

在 Oracle Solaris 11.2 之前的发行版中，虚拟交换机是通过创建 VNIC 时所基于的数据链路间接管理的。从 Oracle Solaris 11.2 发行版开始，可通过 EVS 管理虚拟交换机。您可以显式创建一个虚拟交换机并指定名称，为该虚拟交换机指定虚拟端口 (virtual ports, VPort)，并将其与 IP 地址块相关联。可以为虚拟端口设置属性，例如优先级、最大带宽、服务类 (class of service, CoS)、MAC 地址和 IP 地址。您还可以以虚拟交换机为单位配置缺省的 SLA。

注 - 在 VNIC 创建过程中隐式创建的虚拟交换机将继续存在，并且在此发行版中的作用与以前发行版中的作用相同。EVS 未替代现有的隐式虚拟交换机。

下图显示了单个计算节点中的弹性虚拟交换机 EVS0。

图 5-1 计算节点中的弹性虚拟交换机



Oracle Solaris 弹性虚拟交换机的功能是什么？

使用 Oracle Solaris 弹性虚拟交换机 (Elastic Virtual Switch, EVS) 功能，可以创建和管理跨一个或多个计算节点的虚拟交换机。这些计算节点是托管 VM 的物理计算机。弹性虚拟交换机是一个实体，表示属于同一第 2 层 (Layer 2, L2) 网段的显式创建的虚拟交换机。弹性虚拟交换机在从网络中任意位置连接到它的 VM 之间提供网络连接。

注 - 在 EVS 中，所有提及虚拟机 (virtual machine, VM) 一词的地方特指 Oracle Solaris 区域和 Oracle Solaris 内核区域。

弹性虚拟交换机可跨越多个主机。这些虚拟交换机被称为“弹性”是因为它们可以跨入主机，也可以跨出主机。在将主机的 VNIC 连接到弹性虚拟交换机时，弹性虚拟交换机将跨入主机。删除这些 VNIC 后，弹性虚拟交换机将跨出主机。

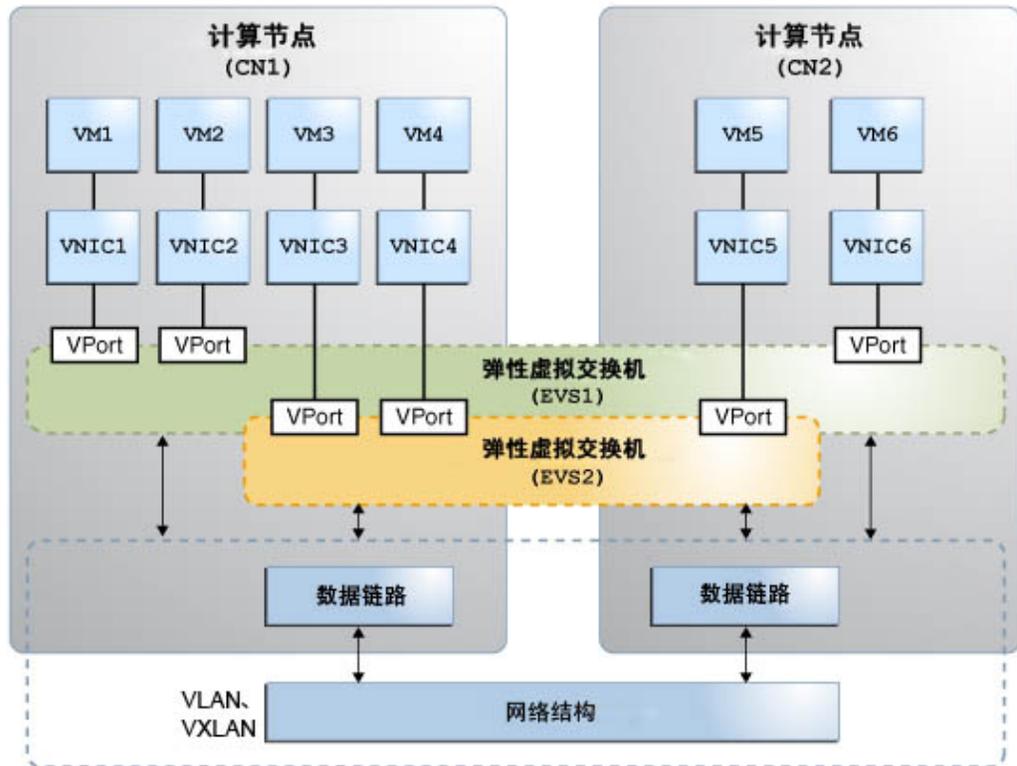
弹性虚拟交换机表示一个隔离的 L2 网段，该隔离是通过 VLAN 或 VXLAN 实现的。有关如何使用 VLAN 实现弹性虚拟交换机的信息，请参见[“使用案例：配置弹性虚拟交换机” \[130\]](#)。有关如何使用 VXLAN 实现弹性虚拟交换机的信息，请参见[“使用案例：为租户配置弹性虚拟交换机” \[136\]](#)。

有关管理 VLAN 的信息，请参见《[在 Oracle Solaris 11.2 中管理网络数据链路](#)》中的第 3 章“[使用虚拟局域网配置虚拟网络](#)”。有关管理 VXLAN 的信息，请参见第 3 章 [使用虚拟可扩展局域网配置虚拟网络](#)。

每个弹性虚拟交换机都与一个名称、多个虚拟端口和一个 IP 地址块相关联。您可以创建、监视和控制虚拟交换机资源。有关更多信息，请参见第 6 章 [管理弹性虚拟交换机](#)。

下图显示了两个计算节点之间的两个弹性虚拟交换机 (EVS1 和 EVS2)。在这些计算节点上置备的 VM 是通过跨越这两个计算节点的弹性虚拟交换机连接的。每个计算节点通过一个数据链路连接到同一网络结构。数据链路也称为上行链路端口。这些计算节点上的数据链路将虚拟交换机连接到外部网络。VNIC 通过虚拟端口 (virtual port, VPort) 连接到弹性虚拟交换机。VNIC 继承与虚拟端口关联的属性，例如 MAC 地址、IP 地址和 SLA。

图 5-2 计算节点之间的弹性虚拟交换机



在此图中，VM VM1、VM2 和 VM6 可以通过弹性虚拟交换机 EVS1 相互通信。VM VM3、VM4 和 VM5 可以通过弹性虚拟交换机 EVS2 相互通信。有关更多信息，请参见[如何配置弹性虚拟交换机 \[114\]](#)。

使用 EVS 的益处

在托管着若干个虚拟机的数据中心环境中，EVS 提供了以下益处，从而使某些网络管理任务变得更加简单：

- 在位于多个服务器上的 VM 之间创建虚拟网络，从而提供网络连接
- 支持添加具有定制 SLA 的虚拟端口
- 通过使用 VLAN 或 VXLAN 来提供网络隔离
- 支持共享同一底层基础结构的多租户虚拟网络

- 与 Oracle Solaris 区域和 Oracle Solaris 内核区域集成
- 提供以下项的集中管理：
 - 虚拟端口的 MAC 地址和 IP 地址
 - 每个虚拟交换机或每个虚拟端口的 SLA
 - 监视虚拟端口的运行时网络通信统计数据

弹性虚拟交换机资源

弹性虚拟交换机与下列主要资源关联：IP 网络和虚拟端口。

IP 网络

IP 网络也称为 IPnet，表示一个 IPv4 或 IPv6 地址块，并且此地址块有缺省路由器。此 IPv4 或 IPv6 地址块也称为子网。只能将一个 IPnet 关联到弹性虚拟交换机。对于通过虚拟端口连接到弹性虚拟交换机的所有 VM，系统都会指定一个与弹性虚拟交换机关联的 IPnet 中的 IP 地址。

您还可以通过为 VPort 设置 IP 地址属性 `ipaddr`，手动将 IP 地址指定给 VM。此 IP 地址必须位于 IPnet 的子网范围内。有关如何将 IPnet 添加到弹性虚拟交换机的更多信息，请参见[如何配置弹性虚拟交换机 \[114\]](#)。

虚拟端口

虚拟端口也称为 VPort，表示 VNIC 和弹性虚拟交换机之间的连接点。当 VNIC 连接到 VPort 时，VNIC 将继承 VPort 封装的网络配置参数，例如：

- SLA 参数，例如，最大带宽、服务类和优先级
- MAC 地址
- IP 地址

创建 VPort 时，随机生成的 MAC 地址以及关联的 IPnet 中的下一可用 IP 地址将分配到 VPort。随机生成的 MAC 地址具有缺省前缀，包含有效 IEEE OUI 和本地位设置。您还可以在通过使用 `evsadm add-vport` 命令添加 VPort 时指定 IP 地址和 MAC 地址。有关如何添加 VPort 的更多信息，请参见[如何配置弹性虚拟交换机 \[114\]](#)。

注 - 您无需始终将虚拟端口添加到弹性虚拟交换机。创建 VNIC 时，您只能指定 VNIC 必须连接到的弹性虚拟交换机的名称。在这种情况下，EVS 控制器将生成系统虚拟端口。这些虚拟端口遵循命名约定 `sys-vportname`，例如 `sys-vport0`。系统虚拟端口继承弹性虚拟交换机的属性。

下表显示了 VPort 属性。

表 5-1 VPort 属性

VPort 属性	说明	可能的值	缺省值
cos	为 VPort 上的传出包指定 802.1p 优先级。	0 - 7	--
maxbw	为 VPort 指定全双工带宽。	--	--
priority	为 VPort 指定相对优先级。	high、medium 或 low	medium
ipaddr	指定与虚拟端口关联的 IP 地址。只能在创建 VPort 时指定 IP 地址。	--	如果没有为 VPort 指定 IP 地址，则 EVS 控制器将自动从与弹性虚拟交换机关联的 IPnet 中选择 IP 地址。
macaddr	指定与 VPort 关联的 MAC 地址。只能在创建 VPort 时指定 MAC 地址。	--	如果没有为 VPort 指定 MAC 地址，则 EVS 控制器将为 VPort 生成随机的 MAC 地址。
evs	一个只读属性，表示 VPort 所关联的弹性虚拟交换机。	--	--
tenant	一个只读属性，表示 VPort 与之关联的 tenant。	--	--

不能修改属性 `evs` 和 `tenant`，因为它们是只读属性。有关 VPort 属性的更多信息，请参见 [evsadm\(1M\)](#) 手册页。

EVS 中的名称空间管理

弹性虚拟交换机及其资源从逻辑上分组在一起。每个逻辑组称为一个租户。在一个租户内定义的弹性虚拟交换机的资源在该租户的名称空间外部不可见。租户用作一个容器，将该租户的所有资源存放在一起。有关如何使用租户创建弹性虚拟交换机的更多信息，请参见[如何配置弹性虚拟交换机 \[114\]](#)。

您不需要为任何 EVS 操作指定租户名称。缺省租户名称为 `sys-global`，所有 EVS 操作都将在此名称空间中进行。

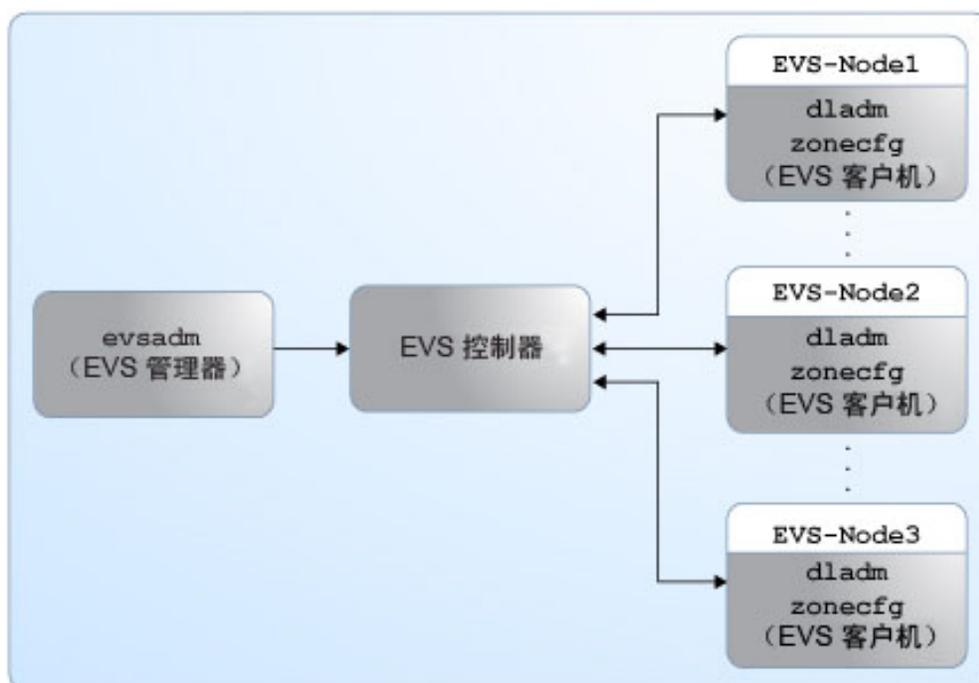
EVS 组件

EVS 包含以下组件：

- EVS 管理器
- EVS 控制器
- EVS 客户机
- EVS 节点

下图显示了 EVS 的组件。

图 5-3 EVS 组件



在下图中，EVS 管理器和 EVS 控制器是两个单独的主机。EVS 节点 EVS-Node1、EVS-Node2 和 EVS-Node3 是三个主机，其 VNIC 或区域的 VNIC anet 资源连接到弹性虚拟交换机。

EVS 管理器

EVS 管理器是与 EVS 控制器进行通信来定义 L2 网络拓扑和必须在这些 L2 网络上使用的 IP 地址的实体。EVS 管理器通过使用 `evsadm` 命令与 EVS 控制器进行通信。EVS 管理器和 EVS 控制器也可以位于同一个计算节点上。

注 - L2 网络拓扑是网段，并且每个网段形成单个广播域，这是通过使用 VLAN 或 VXLAN 实现的。

在安装 `service/network/evs` 软件包后，您可以在 EVS 管理器上执行 EVS 操作，并通过结合使用 `controller` 属性和 `evsadm set-prop` 命令来指定 EVS 控制器。指定 `controller` 属性时，请采用 `ssh://[user@]example-controller.com` 格式。有关更多信息，请参见第 6 章 [管理弹性虚拟交换机](#)。

EVS 控制器

EVS 控制器提供配置和管理弹性虚拟交换机及其关联的所有资源的功能。在数据中心中，只能将一个物理计算机设置为 EVS 控制器。

可通过结合使用 `controller` 属性和 `evsadm set-prop` 命令来指定 EVS 控制器。`controller` 属性保存在 `svc:/network/evs:default` SMF 服务中，因此在系统引导后也持久存在。

此 EVS 控制器与您可以通过使用 `evsadm set-controlprop` 命令配置的属性相关联。要跨物理计算机实现 L2 网段，您需要为 EVS 控制器的属性配置信息，例如，可用 VLAN ID、可用 VXLAN 网段 ID 或每个 EVS 节点的上行链路端口。有关如何配置 EVS 控制器并为其设置属性的更多信息，请参见“[创建和管理 EVS 控制器](#)” [103]。

注 - 您还可以通过使用 SMF 站点配置文件和自动安装 (Auto Install, AI) 服务，将 EVS 控制器信息推送到数据中心的每个 EVS 节点。有关 SMF 的更多信息，请参见《[在 Oracle Solaris 11.2 中管理系统服务](#)》。有关 AI 服务的更多信息，请参见《[安装 Oracle Solaris 11.2 系统](#)》中的“[使用安装服务](#)”。

下表显示了可为 EVS 控制器配置的属性。

表 5-2 EVS 控制器属性

EVS 控制器属性	说明	可能的值	缺省值
<code>l2-type</code>	定义如何在物理计算机间实现弹性虚拟交换机。	<code>vlan</code> 或 <code>vxlan</code>	<code>vlan</code>

EVS 控制器属性	说明	可能的值	缺省值
	注 - 更改 l2-type 属性时，在更改之前创建的弹性虚拟交换机不受影响。仅在更改之后创建的弹性虚拟交换机才具有更新后的 l2-type 属性。此行为意味着基于 VLAN 和 VXLAN 的 L2 网段可以在 EVS 控制器中共存。		
vlan-range	用于创建弹性虚拟交换机的 VLAN ID 范围的逗号分隔列表。每个弹性虚拟交换机都与一个 VLAN ID 相关联。	1 - 4094	--
vxlان-range	用于创建弹性虚拟交换机的 VXLAN 网段编号范围的逗号分隔列表。每个弹性虚拟交换机都与一个 VXLAN 网段编号相关联。	0 - 16777215	--
vxlان-addr	指定创建 VXLAN 数据链路时必须基于的 IP 地址。您也可以将 vxlan-addr 属性设置为子网。	--	--
vxlان-mgroup	指定在创建 VXLAN 数据链路时需要使用的多播地址。	--	如果未指定多播地址，则 VXLAN 数据链路将使用 All Host 地址。
vxlان-ipvers	指定需要用于托管 VXLAN 数据链路的 IP 接口的 IP 版本的地址。	v4 或 v6	v4
uplink-port	指定需要用于 VLAN 或 VXLAN 的数据链路。	--	--

为 EVS 控制器设置的控制器属性适用于整个数据中心。但是，您可以以主机为单位覆盖控制器属性 uplink-port 和 vxlan-addr 的值。

例如，假定您在设置控制器属性时，将 uplink-port 属性设置到数据链路 net2，用于在数据中心中的每个 EVS 节点上创建 VNIC 或 VXLAN。但是，如果数据中心中的 EVS 节点具有数据链路 net1 作为唯一接口，将需要使用每主机值覆盖全局值 net2，如下所示：

```
# evsadm set-controlprop -h host1 -p uplink-port=net1
```

有关更多信息，请参见[如何配置 EVS 控制器 \[110\]](#)。

如果未指定控制器属性的值，则此属性将重置为缺省值，如[例 6-2 “重置 EVS 控制器的属性”](#) 中所示。有关 EVS 控制器属性的更多信息，请参见 [evsadm\(1M\)](#) 手册页。

EVS 客户机

`dladm` 和 `zonecfg` 命令是 EVS 客户机。通过使用弹性虚拟交换机、IPnet 和 VPort，可以通过 `evsadm` 命令定义 L2 网络拓扑。您可以使用 `dladm` 命令将 VNIC 连接到 L2 网络拓扑，或者使用 `zonecfg` 命令连接 VNIC anet 资源，从而将区域连接到 L2 网络拓扑。

注 - `evsadm` 命令是用于定义 L2 网络拓扑的 EVS 管理器。

通过使用 `dladm` 命令或 `zonecfg` 命令为弹性虚拟交换机创建 VNIC 时，将从 EVS 控制器检索 VNIC 的配置信息。

在安装 `service/network/evs` 软件包后，您可以在 EVS 客户机上执行 EVS 操作，并通过结合使用 `controller` 属性和 `evsadm set-prop` 命令来指定 EVS 控制器。指定 `controller` 属性时，请采用 `ssh://[user@]example-controller.com` 格式。有关更多信息，请参见 [第 6 章 管理弹性虚拟交换机](#)。

EVS 节点

EVS 节点是其 VNIC 或区域的 VNIC anet 资源连接到弹性虚拟交换机的主机。可以使用诸如 `dladm` 和 `zonecfg` 之类的命令指定需要连接到弹性虚拟交换机的 VNIC。有关更多信息，请参见“[为弹性虚拟交换机创建 VNIC](#)” [115]。

EVS 管理命令

可以使用下面的管理命令来管理弹性虚拟交换机：

- `evsadm`
- `evsstat`
- `dladm`
- `zonecfg`

有关如何配置弹性虚拟交换机的信息，请参见[如何配置弹性虚拟交换机](#) [114]。

evsadm 命令

您使用 `evsadm` 命令与 EVS 控制器通信并管理弹性虚拟交换机、IPnet 和 VPort。本节介绍了与此命令一起执行活动的子命令。有关更多信息，请参见 [evsadm\(1M\)](#) 手册页。

用于管理弹性虚拟交换机的 `evsadm` 子命令

用于管理虚拟交换机的 `evsadm` 子命令包括：

<code>create-evs</code>	创建弹性虚拟交换机
<code>delete-evs</code>	删除弹性虚拟交换机
<code>show-evs</code>	显示有关弹性虚拟交换机的信息
<code>set-evsprop</code>	允许设置弹性交换机的 <code>maxbw</code> 和 <code>priority</code> 属性 有关这些属性的更多信息，请参见 “设置弹性虚拟交换机的属性” [119] 。
<code>show-evsprop</code>	显示弹性虚拟交换机的属性

用于管理 IPnet 的 `evsadm` 子命令

用于管理 IPnet 的 `evsadm` 子命令包括：

<code>add-ipnet</code>	将 IPnet 添加到弹性虚拟交换机并使您能够设置 <code>subnet</code> 和 <code>defrouter</code> 属性 有关这些属性的更多信息，请参见 “将 IPnet 添加到弹性虚拟交换机” [113] 。
<code>remove-ipnet</code>	删除 IPnet
<code>show-ipnet</code>	显示有关 IPnet 的信息

用于管理 VPort 的 `evsadm` 子命令

用于管理虚拟端口的 `evsadm` 子命令包括：

<code>add-vport</code>	添加 VPort
<code>remove-vport</code>	删除 VPort
<code>show-vport</code>	显示有关 VPort 的信息
<code>set-evsprop</code>	使您能够为 VPort 设置以下属性： <ul style="list-style-type: none"> ■ <code>cos</code>

- maxbw
- priority

有关这些属性的更多信息，请参见表 5-1 “VPort 属性”。

show-vportprop	显示 VPort 的属性
reset-vport	重置 VPort

用于管理 EVS 客户机属性的 evsadm 子命令

用于管理 EVS 客户机属性的 evsadm 子命令包括：

set-prop	使您能够设置 controller 属性
show-prop	显示 EVS 客户机属性

用于管理 EVS 控制器属性的 evsadm 子命令

用于管理 EVS 控制器属性的 evsadm 子命令包括：

set-controlprop	使您能够为控制器设置以下属性：
	<ul style="list-style-type: none">▪ l2-type▪ vlan-range▪ vxlan-range▪ vxlan-mgroup▪ vxlan-addr▪ vxlan-ipvers▪ uplink-port

有关这些属性的更多信息，请参见表 5-2 “EVS 控制器属性”。

show-controlprop	显示 EVS 控制器的属性
------------------	---------------

evsstat 命令

evsstat 命令可显示数据中心的所有 VPort 或指定弹性虚拟交换机的所有 VPort 的网络通信统计数据。它还报告与 VPort 关联的 VNIC 的统计信息。有关更多信息，请参见[“监](#)

视弹性虚拟交换机” [128]。有关 `evsstat` 命令的更多信息，请参见 [evsstat\(1M\)](#) 手册页。

dladm 命令

可以使用下面的 `dladm` 命令来管理连接到弹性虚拟交换机的 VNIC：

- `dladm create-vnic` 命令 – 用于创建 VNIC 并指定您需要将 VNIC 连接到的弹性虚拟交换机的名称。您还可以指定弹性虚拟交换机的 VPort。
- `dladm show-vnic` 命令 – 用于显示特定 VNIC 的弹性虚拟交换机信息。`dladm show-vnic` 命令的输出还显示字段 `TENANT`、`EVS` 和 `VPORT`。不过，这些字段在区域内不可见。

有关更多信息，请参见 [dladm\(1M\)](#) 手册页。

有关如何为弹性虚拟交换机配置 VNIC 的更多信息，请参见[如何为弹性虚拟交换机创建 VNIC \[116\]](#)。

zonecfg 命令

使用增强的 `zonecfg` 命令为弹性虚拟交换机配置区域的 VNIC `anet` 资源。可以为 VNIC `anet` 资源设置以下属性：

- `tenant` – 指定租户的名称。如果在配置区域时未指定值，则系统将指定缺省值 `sys-global`。
- `vport` – 指定 VPort 的名称。如果在配置区域时未指定值，则系统将为弹性虚拟交换机生成 VPort，并且 VPort 将继承弹性虚拟交换机的属性。
- `evs` – 指定必须将 VNIC `anet` 资源连接到的弹性虚拟交换机的名称。

有关 `anet` 资源的更多信息，请参见《[Oracle Solaris Zones 介绍](#)》中的“资源类型属性”中的 `anet` 描述。

注 - 区域配置必须包括租户名称、弹性虚拟交换机名称和用于唯一标识数据中心的 VPort 的 VPort 名称。有关区域配置的更多信息，请参见《[创建和使用 Oracle Solaris 区域](#)》。

有关如何为弹性虚拟交换机配置 VNIC `anet` 资源的更多信息，请参见“[为弹性虚拟交换机创建 VNIC anet 资源](#)” [116]。有关 `zonecfg` 命令的更多信息，请参见 [zonecfg\(1M\)](#) 手册页。

有关管理连接到弹性虚拟交换机的 VNIC 的限制

以下限制适用于您使用 `dladm create-vnic` 命令或 `zonecfg` 命令创建并连接到弹性虚拟交换机的 VNIC：

- 无法使用 `dladm rename-link` 命令重命名 VNIC。
- 无法使用 `dladm set-linkprop` 或 `dladm reset-linkprop` 命令更改此类 VNIC 的属性。
- 无法使用 `dladm modify-vnic` 命令修改这些 VNIC。

自动生成的 VXLAN 数据链路

如果您使用 VXLAN 为弹性虚拟交换机实现第 2 层网段，则 EVS 将自动在托管弹性虚拟交换机的 VNIC 的 EVS 节点上创建 VXLAN 数据链路。这些数据链路称为自动生成的 VXLAN 数据链路，并遵循命名约定 `evs-vxlansegment-ID`，其中 `evs` 是创建了此数据链路的实体。例如，名称 `evs-vxlan200` 指示 200 为 VXLAN ID，`evs` 是已创建此数据链路的实体。您可以使用 `dladm show-vxlan` 命令显示自动生成的 VXLAN 数据链路。有关更多信息，请参见“[显示 VXLAN 信息](#)” [61]。

对于自动生成的 VXLAN 数据链路，无法使用 `dladm` 子命令来删除或重命名数据链路。但是，您可以使用 `dladm set-linkprop` 命令和 `dladm reset-linkprop` 命令来临时设置数据链路属性。

使用 EVS 时必需的软件包

在使用 EVS 之前，您需要安装以下软件包：

- `pkg:/service/network/evs`
您需要在 EVS 管理器、EVS 控制器和 EVS 节点上安装核心软件包 `pkg:/service/network/evs`。此软件包中包含以下组件：
 - `evsadm`
 - `evsstat`
 - SMF 服务 (`svc:/network/evs:default`) – 此 SMF 服务具有用于保存 EVS 控制器的主机名或 IP 地址的 `controller` 属性。EVS 客户机使用主机名或 IP 地址与 EVS 控制器通信。可以使用 `evsadm set-prop` 命令管理 `controller` 属性。

安装 `pkg:/service/network/evs` 软件包时，将创建新用户 `evsuser`。`evsuser` 是具有“弹性虚拟交换机管理”权限配置文件的特定用户。此配置文件提供用于执行 EVS 操作的所有授权和特权。

- `pkg:/system/management/rad/module/rad-evs-controller`
 您只需要在用作 EVS 控制器的系统上安装此软件包。必须只使用一个控制器来管理数据中心的所有弹性虚拟交换机。此软件包中包含 SMF 服务 `svc:/network/evs-controller:default`。此 SMF 服务包含的属性用于捕获在物理计算机间实现 L2 网段时所需的信息。可以使用 `evsadm set-controlprop` 命令管理控制器属性。

有关更多信息，请参见“[EVS 控制器的必需软件包](#)” [103]。

EVS 如何使用区域

通过使用与 `zonecfg` 命令关联的属性，可以将 VNIC `anet` 资源连接到弹性虚拟交换机。Oracle Solaris 区域和 Oracle Solaris 内核区域支持 EVS 功能。

内核区域支持为弹性虚拟交换机创建的 VNIC。仅当在内核区域内部创建的 VNIC 使用与 `zvnet` 驱动程序关联的出厂 MAC 地址时，该 VNIC 才起作用。由于为弹性虚拟交换机创建的 VNIC 会继承与弹性虚拟交换机的 VPort 关联的 MAC 地址，因此必须通过将 `macaddr` 属性设置为 `zvnet` 驱动程序的出厂 MAC 地址来为弹性虚拟交换机创建 VPort。

可以使用下面的命令语法显式指定出厂 MAC 地址：

```
# evsadm add-vport -p macaddr=factory-MAC-addr-zvnet EVS-name/VPort-name
```

在内核区域中，可以将 VNIC 连接到使用此命令创建的 VPort。有关内核区域的信息，请参见《[创建和使用 Oracle Solaris 内核区域](#)》。

使用 EVS 时的安全要求

要执行 EVS 操作，您需要成为超级用户，或是具有“弹性虚拟交换机管理”权限配置文件的用户。您也可以创建一个用户，然后将“弹性虚拟交换机管理”权限配置文件指定给该用户。有关更多信息，请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》。

注 - 在多租户 EVS 设置中，单个租户不能管理自己的弹性虚拟交换机及其资源，因为不支持对每个用户采用租户模式的用户授权。整个 EVS 域只能由一个管理员来管理所有租户的资源。

以下示例显示了如何创建具有“弹性虚拟交换机管理”权限配置文件的 `user1`。

```
# useradd -P "Elastic Virtual Switch Administration" user1
```

以下示例显示了如何向现有用户 `user1` 添加“弹性虚拟交换机管理”权限配置文件。

```
# usermod -P +"Elastic Virtual Switch Administration" user1
```

设置 EVS 控制器时，必须指定具有“弹性虚拟交换机管理”权限配置文件的用户。例如，在设置 EVS 控制器时必须指定 user1，如下所示：

```
# evsadm set-prop -p controller=ssh://user1@example-controller.com
```

有关更多信息，请参见[“配置 EVS 控制器” \[105\]](#)。

注 - 也可以使用在安装 pkg:/service/network/evs 软件包时创建的 evsuser。用户 evsuser 指定有“弹性虚拟交换机管理”权限配置文件。此配置文件提供用于执行 EVS 操作的所有授权和特权。

管理弹性虚拟交换机

本章介绍了用于管理弹性虚拟交换机及其资源的任务。有关一般信息，请参见[第 5 章 关于弹性虚拟交换机](#)。

本章包含以下主题：

- “EVS 管理任务” [101]
- “规划弹性虚拟交换机配置” [102]
- “创建和管理 EVS 控制器” [103]
- “配置弹性虚拟交换机” [112]
- “管理弹性虚拟交换机、IPnet 和 VPort” [117]
- “监视弹性虚拟交换机” [128]
- “弹性虚拟交换机的使用案例示例” [130]

EVS 管理任务

本节提供用于完成 EVS 管理任务的下列信息：

- 如何配置 EVS 控制器 [110]
- 如何配置弹性虚拟交换机 [114]
- “为弹性虚拟交换机创建 VNIC” [115]
- “显示弹性虚拟交换机信息” [118]
- “设置弹性虚拟交换机的属性” [119]
- “显示弹性虚拟交换机的属性” [120]
- “删除 IPnet” [121]
- “显示 IPnet” [122]
- “设置 VPort 的属性” [123]
- “显示 VPort 的属性” [124]
- “显示 VPort” [125]
- “删除 VPort” [126]

- [如何删除弹性虚拟交换机 \[127\]](#)
- [“监视弹性虚拟交换机” \[128\]](#)

规划弹性虚拟交换机配置

规划弹性虚拟交换机配置包括以下操作：

1. 在 EVS 控制器、EVS 管理器和 EVS 节点上安装必需的软件包。必须单独为其中每个组件安装这些软件包。有关更多信息，请参见[“使用 EVS 时必需的软件包” \[98\]](#)。
2. 在 EVS 设置中，使用在以下组件之间为 evsuser 预先共享的公钥来设置 SSH 验证：
 - EVS 管理器和 EVS 控制器
 - 每个 EVS 节点和 EVS 控制器
 - EVS 控制器和每个 EVS 节点

有关更多信息，请参见[“设置 SSH 验证” \[106\]](#)。

3. 通过设置 controller 属性指定 EVS 控制器。必须在 EVS 节点、EVS 管理器和 EVS 控制器上指定 EVS 控制器的主机名或 IP 地址。有关更多信息，请参见[“配置 EVS 控制器” \[105\]](#)。
4. 配置 EVS 控制器，这涉及：
 - a. 设置 EVS 控制器的属性。
 - b. 验证为 EVS 控制器设置的属性。

有关更多信息，请参见[如何配置 EVS 控制器 \[110\]](#)。

5. 通过使用 EVS 管理器来配置弹性虚拟交换机，其中涉及：
 - a. 创建弹性虚拟交换机。
 - b. 将 IPnet 添加到弹性虚拟交换机。
 - c. 将 VPort 添加到弹性虚拟交换机。
 - d. 验证配置的弹性虚拟交换机。

有关更多信息，请参见[如何配置弹性虚拟交换机 \[114\]](#)。

6. 在 EVS 节点上创建 VNIC 并将 VNIC 连接到弹性虚拟交换机，这涉及：
 - a. 使用 dladm 命令创建 VNIC，或者使用 zonecfg 命令创建 VNIC anet 资源并将其连接到弹性虚拟交换机。
 - b. 验证连接到弹性虚拟交换机的 VNIC。

有关更多信息，请参见[“为弹性虚拟交换机创建 VNIC” \[115\]](#)。

创建和管理 EVS 控制器

EVS 控制器提供了用于配置和管理弹性虚拟交换机及其关联的所有资源的功能。必须设置 EVS 控制器的属性，该控制器捕获在物理计算机中实现第 2 层网段所需的信息。有关更多信息，请参见“EVS 控制器” [92]。

规划 EVS 控制器包括以下注意事项：

- 确定是通过 VLAN、VXLAN 还是这两者来实现弹性虚拟交换机。
 - 如果使用 VLAN 实现弹性虚拟交换机，则需要设置属性 `uplink-port` 和 `vlan-range`。
 - 如果使用 VXLAN 实现弹性虚拟交换机，则需要设置属性 `vxlan-range` 和 `uplink-port` 或 `vxlan-addr`。您还可以设置属性 `vxlan-mgroup` 和 `vxlan-ipvers`。

注 - 创建弹性虚拟交换机后，不能修改该弹性虚拟交换机的 EVS 控制器属性。对 EVS 控制器属性进行的任何修改将反映在所创建的新的弹性虚拟交换机中。

- 如果计算节点没有相同的数据链路，则针对每个计算节点，您需要为 `uplink-port` 属性指定数据链路。

例如，假定有两个计算节点，具有数据链路 `net2` 的 `host1` 以及具有数据链路 `net3` 的 `host2`。在设置 `uplink-port` 属性时，需要同时指定两个主机的数据链路，如下所示：

```
# evsadm set-controlprop -h host1 -p uplink-port=net2
# evsadm set-controlprop -h host2 -p uplink-port=net3
```

EVS 控制器的必需软件包

必须只使用一个控制器来管理数据中心的所有弹性虚拟交换机。必须在充当 EVS 控制器的系统上安装 `pkg:/service/network/evs` 软件包和 `pkg:/system/management/rad/module/rad-evs-controller` 软件包。

使用以下命令安装这些软件包：

```
# pkg install evs
# pkg install rad-evs-controller
```

安装 `rad-evs-controller` 软件包后，您需要使用以下命令重新启动 `rad:local` 服务来加载 EVS 控制器：

```
# svcadm restart rad:local
```

用于配置 EVS 控制器的命令

本节说明了如何对 EVS 控制器执行以下任务：

- 设置 EVS 控制器
- 显示 EVS 控制器
- 设置 EVS 控制器的属性
- 显示 EVS 控制器的属性

设置 EVS 控制器

可以使用 `evsadm set-prop` 命令在主机上设置 EVS 控制器。命令语法为：

```
# evsadm set-prop -p controller=[value[...]]
```

此命令为执行此命令的主机设置属性值。唯一支持的属性是 `controller`，格式可以是：`ssh://[user@]evs-controller-host-name` 或 `ssh://[user@]evs-controller-IP-address`。

显示 EVS 控制器

可以使用 `evsadm show-prop` 命令显示 EVS 控制器。命令语法为：

```
# evsadm show-prop [[-c] -o field[,...]] [-p controller[,...]]
```

`-p controller` 指定 RAD 客户机必须连接到的 EVS 控制器。

`-o field[,...]` 指定要显示的输出字段的逗号分隔列表，不区分大小写。可以指定以下字段，它们在输出中显示为列：

<code>all</code>	显示所有输出字段
<code>PROPERTY</code>	属性的名称
<code>PERM</code>	属性的权限，为 <code>rw</code> 或 <code>r-</code>
<code>VALUE</code>	属性的值
<code>DEFAULT</code>	属性的缺省值

`-c` 使用稳定的可供计算机解析的格式显示。需要同时指定 `-o` 选项和 `-c` 选项。

有关说明如何显示 EVS 控制器的示例，请参见[例 6-1 “配置 EVS 控制器”](#)。

设置 EVS 控制器的属性

可以使用 `evsadm set-controlprop` 命令设置 EVS 控制器的属性。命令语法为：

```
# evsadm set-controlprop [-h host] -p prop=[value[...]]
```

- h *host* 指定为其设置属性的主机。
- p *prop* 指定为 EVS 控制器设置的控制器属性的名称。如果属性接受多个值，则必须使用逗号作为分隔符来指定这些值。一次只能指定一个属性。如果未指定值，则属性将重置为缺省值。有关可为 EVS 控制器设置的属性的更多信息，请参见表 5-2 “EVS 控制器属性”。

显示 EVS 控制器的属性

可以使用 `evsadm show-controlprop` 命令显示 EVS 控制器的属性。命令语法为：

```
# evsadm show-controlprop [[-c] -o field[,...]] [-p prop[,...]]
```

此命令可显示 EVS 控制器的一个或多个属性的当前值。如果没有为 EVS 控制器指定属性，则将显示控制器的所有现有属性。有关控制器属性的更多信息，请参见表 5-2 “EVS 控制器属性”。

- o *field[,...]* 指定要显示的输出字段的逗号分隔列表，不区分大小写。可以指定以下字段，它们在输出中显示为列：

all	显示所有输出字段。
PROPERTY	属性的名称。
PERM	属性的权限，为 <code>rw</code> 或 <code>r-</code> 。
VALUE	属性的值。
DEFAULT	属性的缺省值。
HOST	如果值为 <code>--</code> ，则表示该属性是全局的，适用于所有主机。否则，该属性适用于特定主机。

有关说明如何显示 EVS 控制器属性的示例，请参见例 6-1 “配置 EVS 控制器”。

配置 EVS 控制器

必须只配置一个计算节点作为网络中的 EVS 控制器，然后在每个 EVS 节点上设置 EVS 控制器，以便 EVS 节点可以与 EVS 控制器通信。但是，您只需要从可与 EVS 控制器

通信的任何节点设置一次 EVS 控制器的属性。可以使用 `evsadm set-controlprop` 命令设置 EVS 控制器的属性。有关更多信息，请参见[如何配置 EVS 控制器 \[110\]](#)。

也可以重置 EVS 控制器的属性。[例 6-2 “重置 EVS 控制器的属性”](#) 说明了如何重置 EVS 控制器的属性。有关 EVS 控制器及其属性的信息，请参见[“EVS 控制器” \[92\]](#)。

要简化弹性虚拟交换机的配置，您需要以 `evsuser` 身份进行连接。在安装必需的 EVS 软件包 (`service/network/evs`) 时，将创建一个特殊用户 `evsuser`，并向其指定“弹性虚拟交换机管理”权限配置文件。此配置文件包含执行 EVS 操作的所有授权和特权。要使用 `evsuser`，您需要按如下所示设置 `controller` 属性：

```
# evsadm set-prop -p controller=ssh://evsuser@evs-controller-hostname-or-IP-address
```

此外，您还必须使用在运行 `evsadm` 命令的主机与 EVS 控制器之间预先共享的公钥来设置 SSH 验证。

注 - 要执行 EVS 操作，您需要是超级用户，或者具有“弹性虚拟交换机管理”权限配置文件的用户。有关更多信息，请参见[“使用 EVS 时的安全要求” \[99\]](#)。

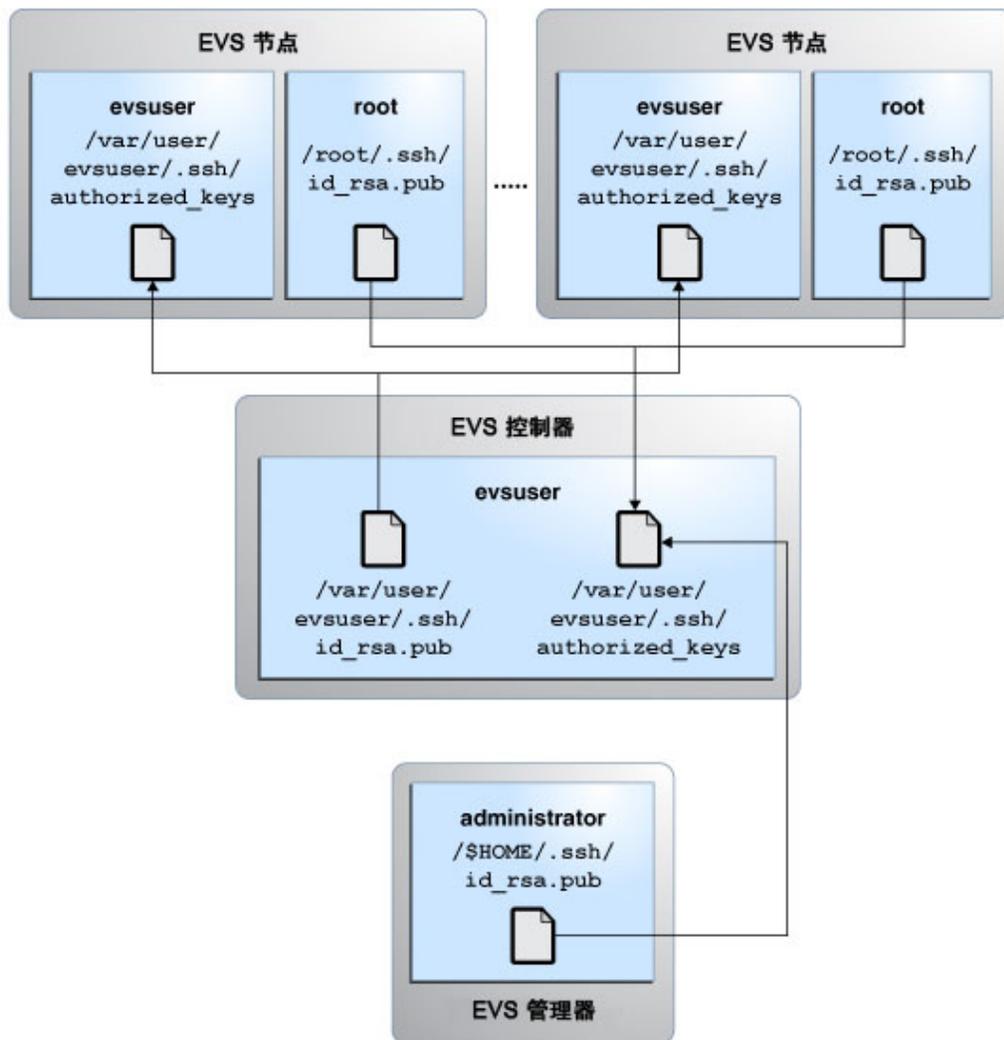
设置 SSH 验证

您需要采用具有预先共享的公钥的 SSH 验证，使 `evsadm` 命令能够以非交互模式安全地与 EVS 控制器通信。在 EVS 设置中，需要使用在以下组件之间为 `evsuser` 预先共享的公钥来设置 SSH 验证：

- EVS 管理器和 EVS 控制器 - 将在 EVS 管理器上运行 `evsadm` 命令的管理员或用户的公钥附加到 EVS 控制器上的 `/var/user/evsuser/.ssh/authorized_keys` 文件中。
- EVS 节点和 EVS 控制器 - 将每个 EVS 节点上的 `root` 用户的公钥附加到 EVS 控制器上的 `/var/user/evsuser/.ssh/authorized_keys` 文件中。需要附加这些公钥的原因是 `zoneadmd` 守护进程以 `root` 用户的身份运行。该守护进程连接到 EVS 控制器，并检索 VNIC `anet` 资源的配置信息。有关更多信息，请参见 [zoneadmd\(1M\)](#) 手册页。
- EVS 控制器和 EVS 节点 - 将 EVS 控制器上的 `evsuser` 的公钥附加到每个 EVS 节点上的 `/var/user/evsuser/.ssh/authorized_keys` 文件，因为 EVS 控制器将与每个 EVS 节点进行通信来设置 `VPort` 属性。

下图显示了设置 EVS 组件之间的 SSH 验证。

图 6-1 EVS 设置中的 SSH 验证



设置 SSH 验证后，需要指定 EVS 控制器。假定 `controller` 属性设置为 EVS 节点、EVS 管理器和 EVS 控制器上的 `ssh://evsuser@evs-controller.example.com`。

以下过程说明如何设置 SSH 验证。

▼ 如何在 EVS 节点和 EVS 控制器之间设置 SSH 验证

1. 成为管理员。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 在 EVS 节点中生成一个 RSA 密钥对。

```
evs-node# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1d:17:04:7d:bf root@evs-node
```

3. 将公钥从 EVS 节点中的 `/root/.ssh/id_rsa.pub` 文件复制到 EVS 控制器中的 `/var/user/evsuser/.ssh/authorized_keys` 文件。

4. 从 EVS 节点以 `evsuser` 的身份登录 EVS 控制器，以验证是否设置了 SSH 验证。

```
evs-node# ssh evsuser@evs-controller
The authenticity of host 'evs-controller (192.168.100.10)' can't be established.
RSA key fingerprint is 73:66:81:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-controller' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:36:28 2014 from evs-controller
Oracle Corporation      SunOS 5.11      11.2      April 2014
evsuser@evs-controllers$
```

该输出显示您可以从 EVS 节点以 `evsuser` 身份登录 EVS 控制器，而不需要使用口令。

▼ 如何在 EVS 管理器和 EVS 控制器之间设置 SSH 验证

1. 成为管理员。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 在 EVS 管理器中生成一个 RSA 密钥对。

```
evs-manager# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
```

```
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1d:17:04:7e:bf root@evs-manager
```

3. 将公钥从 EVS 管理器中的 `/root/.ssh/id_rsa.pub` 文件复制到 EVS 控制器中的 `/var/user/evsuser/.ssh/authorized_keys` 文件。
4. 从 EVS 管理器中以 `evsuser` 身份登录 EVS 控制器，以验证是否设置了 SSH 验证。

```
evs-manager# ssh evsuser@evs-controller
The authenticity of host 'evs-controller (192.168.100.10)' can't be established.
RSA key fingerprint is 73:66:81:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-controller' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:38:28 2014 from evs-controller
Oracle Corporation      SunOS 5.11      11.2      April 2014
evsuser@evs-controller$
```

该输出显示您可以从 EVS 管理器以 `evsuser` 身份登录 EVS 控制器，而不需要使用口令。

▼ 如何在 EVS 控制器和 EVS 节点之间设置 SSH 验证

1. 成为管理员。
有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 成为 EVS 控制器中的 `evsuser` 用户。

```
evs-controller# su - evsuser
```

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

3. 在 EVS 控制器中为 `evsuser` 生成一个 RSA 密钥对。

```
evsuser@evs-controller$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/var/user/evsuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/user/evsuser/.ssh/id_rsa.
Your public key has been saved in /var/user/evsuser/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1e:17:04:7d:bf evsuser@evs-controller
```

4. 将公钥从 EVS 控制器中的 `/var/user/evsuser/.ssh/id_rsa.pub` 文件复制到 EVS 节点中的 `/var/user/evsuser/.ssh/authorized_keys` 文件。
5. 从 EVS 控制器以 `evsuser` 身份登录 EVS 节点，以验证是否设置了 SSH 验证。

```
evsuser@evs-controller$ ssh evsuser@evs-node
The authenticity of host 'evs-node (192.168.100.20)' can't be established.
RSA key fingerprint is 73:66:89:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-node' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:40:28 2014 from evs-node
Oracle Corporation      SunOS 5.11      11.2      April 2014
evsuser@evs-node$
```

该输出显示您可以从 EVS 控制器以 evsuser 身份登录 EVS 节点，而不需要使用口令。



注意 - 如果没有在 EVS 设置期间设置 SSH 验证，evsadm 命令将无法以非交互模式安全地与 EVS 控制器通信。

▼ 如何配置 EVS 控制器

开始之前 使用在运行 evsadm 命令的主机与 EVS 控制器之间预先共享的密钥设置 SSH 验证。

1. 成为具有“弹性虚拟交换机管理”权限配置文件的管理员或用户。
有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 设置 EVS 控制器。

```
# evsadm set-prop -p controller=[value[...]]
```

此命令为执行此命令的主机设置属性值。唯一支持的属性是 controller，格式可以是：`ssh://[user@]evs-controller-host-name` 或 `ssh://[user@]evs-controller-IP-address`。

3. (可选) 显示所配置的 EVS 控制器。

```
# evsadm show-prop [[-c] -o field[...]] [-p controller[...]]
```

有关更多信息，请参见“显示 EVS 控制器” [104]。

4. 设置 EVS 控制器的属性。

```
# evsadm set-controlprop [-h host] -p prop=[value[...]]
```

有关更多信息，请参见“设置 EVS 控制器的属性” [105]。

5. (可选) 显示 EVS 控制器的属性。

```
# evsadm show-controlprop [[-c] -o field[...]] [-p prop[...]]
```

有关更多信息，请参见“显示 EVS 控制器的属性” [105]。

例 6-1 配置 EVS 控制器

以下示例说明了如何将主机 s11-server 配置为 EVS 控制器，其 L2 网段是使用 VXLAN 创建的。

```
# evsadm set-prop -p controller=ssh://evsuser@s11-server
# evsadm show-prop
PROPERTY          PERM  VALUE                                DEFAULT
controller         rw   ssh://evsuser@s11-server           --
# evsadm set-controlprop -p l2-type=vxlan
# evsadm set-controlprop -p vxlan-range=10000-20000
# evsadm set-controlprop -p vxlan-addr=192.168.10.0/24
# evsadm set-controlprop -h s11-server -p uplink-port=net3
# evsadm set-controlprop -h s11-client -p uplink-port=net4
# evsadm show-controlprop
PROPERTY          PERM  VALUE                                DEFAULT      HOST
l2-type           rw   vxlan                                vlan         --
uplink-port       rw   net3                                  --           s11-server
uplink-port       rw   net4                                  --           s11-client
vlan-range        rw   --                                    --           --
vlan-range-avail  r-   --                                    --           --
vxlan-addr        rw   192.168.10.0/24                      0.0.0.0     --
vxlan-ipvers      rw   v4                                    v4           --
vxlan-mgroup      rw   0.0.0.0                              0.0.0.0     --
vxlan-range       rw   10000-20000                          --           --
vxlan-range-avail r-   10000-20000                          --           --
```

在此示例中，vxlan-range-avail 属性显示了可用于实现弹性虚拟交换机的 VXLAN ID (10000-20000)。属于子网 192.168.10.0/24 的一个 IP 接口用于在 EVS 节点上创建 VXLAN 链路。

以下示例说明了如何将 IP 地址为 192.168.100.1 的主机配置为 EVS 控制器，其 L2 网段是使用 VLAN 创建的。

```
# evsadm set-prop -p controller=ssh://evsuser@192.168.100.1
# evsadm set-controlprop -p l2-type=vlan
# evsadm set-controlprop -p vlan-range=200-300,400-500
# evsadm set-controlprop -p uplink-port=net2
# evsadm set-controlprop -h host2.example.com -p uplink-port=net3
# evsadm set-controlprop -h host3.example.com -p uplink-port=net4
```

该输出显示为弹性虚拟交换机预留了 VLAN ID 200-300 和 400-500。除了 host2.example.com 和 host3.example.com 之外，所有主机上的数据链路 net2 均为 uplink-port。在 host2 上，数据链路 net3 用作 uplink-port，而在 host3 上，数据链路 net4 用作 uplink-port。

例 6-2 重置 EVS 控制器的属性

以下示例说明了如何重置控制器属性 uplink-port。

```
# evsadm show-controlprop -p uplink-port
```

```

PROPERTY          PERM   VALUE   DEFAULT   HOST
uplink-port       rw     net2    --        --
# evsadm set-controlprop -p uplink-port=
# evsadm show-controlprop -p uplink-port
PROPERTY          PERM   VALUE   DEFAULT   HOST
uplink-port       rw     --      --        --

```

配置弹性虚拟交换机

弹性虚拟交换机是跨一个或多个物理计算机并表示隔离的 L2 网段的虚拟交换机。该隔离通过 VLAN 或 VXLAN 来实现。可以将 EVS 节点的 VNIC 或 anet 资源连接到弹性虚拟交换机，从而在 EVS 节点之间提供网络连接。有关更多信息，请参见[“Oracle Solaris 弹性虚拟交换机的功能是什么？” \[87\]](#)

当您计划配置弹性虚拟交换机时，您需要了解虚拟拓扑。确定需要的 L2 网段数量、每个网络的 IPnet 信息，包括子网和缺省路由器。此外，您可能还希望确定需要为弹性虚拟交换机配置的虚拟端口的数量以及需要为虚拟端口指定的属性。

弹性虚拟交换机的必需软件包

必须在充当 EVS 客户机和 EVS 节点的系统上安装 `pkg:/service/network/evs` 软件包。

使用以下命令安装该软件包：

```
# pkg install evs
```

用于配置弹性虚拟交换机的命令

本节介绍了如何执行以下任务来配置弹性虚拟交换机：

- 创建弹性虚拟交换机
- 将 IPnet 添加到弹性虚拟交换机
- 将 VPort 添加到弹性虚拟交换机

创建弹性虚拟交换机

可以使用 `evsadm create-evs` 命令创建弹性虚拟交换机。命令语法为：

```
# evsadm create-evs [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-name
```

-T *tenant-name* 指定租户。如果指定了某个租户，则将在该租户的名称空间内创建弹性虚拟交换机。否则，将在缺省租户 `sys-global` 中创建弹性虚拟交换机。租户是一个表示弹性虚拟交换机所关联的租户的只读属性。

-p *prop* 指定可在弹性虚拟交换机上设置为指定值的属性的逗号分隔列表。您可以设置以下属性：

- **maxbw** – 为弹性虚拟交换机的端口设置全双工带宽。带宽指定为一个带有容量级后缀（分别表示 Kbps、Mbps 和 Gbps 的 K、M 或 G）的整数。如果未指定单位，输入值将以 Mbps 为单位进行读取。不存在缺省带宽限制。
- **priority** – 为弹性虚拟交换机的端口设置相对优先级。可能的值为 `high`、`medium` 或 `low`。缺省值为 `medium`。此优先级不会反映在线路上的任何协议优先级字段中，而是用于系统内的包处理调度。高优先级 VPort 可能延迟更短，具体取决于可用的系统资源。

EVS-switch-name 指定弹性虚拟交换机的名称。

有关如何创建弹性虚拟交换机的示例，请参见[例 6-3 “配置弹性虚拟交换机”](#)。

将 IPnet 添加到弹性虚拟交换机

可以使用 `evsadm add-ipnet` 命令将 IPnet 添加到弹性虚拟交换机。命令语法为：

```
# evsadm add-ipnet [-T tenant-name] -p subnet=value[{,prop=value[,...]}[,...]]\  
[,...]]\  
EVS-switch-name/IPnet-name
```

-T *tenant-name* 指定租户的名称。如果指定了租户名称，则 IPnet 将与租户名称空间中的 EVS 相关联。

-p *prop* 必须为特定弹性虚拟交换机设置的 IPnet 属性的逗号分隔列表。支持的 IPnet 属性包括：

- **subnet** – 必需。表示 IPv4 或 IPv6 地址的片区。当您添加 IPnet 时，必须指定 `subnet` 属性。否则，添加 IPnet 的操作将失败。
- **defrouter** – 可选。指定给定子网的网关的 IP 地址。未指定 `defrouter` 时，将选择范围中的第一个地址作为缺省的路由器 IP 地址。

EVS-switch-name/IPnet-name 指定具有关联 IPnet 的弹性虚拟交换机的名称。

有关 IPnet 属性的更多信息，请参见 [evsadm\(1M\)](#) 手册页。有关如何将 IPnet 添加到弹性虚拟交换机的示例，请参见 [例 6-3 “配置弹性虚拟交换机”](#)。

将 VPort 添加到弹性虚拟交换机

可以使用 `evsadm add-vport` 命令将 VPort 添加到弹性虚拟交换机。命令语法为：

```
# evsadm add-vport [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-name/VPort-name
```

`-p prop` 指定可为 VPort 设置的 VPort 属性的逗号分隔列表。有关支持的 VPort 属性的更多信息，请参见 [表 5-1 “VPort 属性”](#)。

`EVS-switch-name/VPort-name` 指定具有关联 VPort 的弹性虚拟交换机的名称。

有关如何将 VPort 添加到弹性虚拟交换机的示例，请参见 [例 6-3 “配置弹性虚拟交换机”](#)。

▼ 如何配置弹性虚拟交换机

开始之前 您需要在希望在其上配置弹性虚拟交换机的计算节点上设置 EVS 控制器。有关信息，请参见 [如何配置 EVS 控制器 \[110\]](#) 中的步骤 2。

1. 成为具有“弹性虚拟交换机管理”权限配置文件的管理员或用户。
有关更多信息，请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“[使用所指定的管理权限](#)”。
2. 创建弹性虚拟交换机。

```
# evsadm create-evs [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-name
```

有关更多信息，请参见“[创建弹性虚拟交换机](#)” [112]。

注 - 如果为虚拟端口显式设置了某个属性，则该属性值将覆盖对应的弹性虚拟交换机属性值。

3. 将 IPnet 添加到弹性虚拟交换机。

```
# evsadm add-ipnet [-T tenant-name] -p subnet=value[,{prop=value[,...]}[,...]] \
EVS-switch-name/IPnet-name
```

有关更多信息，请参见“[将 IPnet 添加到弹性虚拟交换机](#)” [113]。

4. (可选) 将 VPort 添加到弹性虚拟交换机。

```
# evsadm add-vport [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-
name/VPort-name
```

将 VPort 添加到弹性虚拟交换机后，将为其指定一个随机 MAC 地址和 IPnet 地址范围中的一个 IP 地址。因此，必须先将 IPnet 添加到弹性虚拟交换机，然后再添加 VPort。有关 `evsadm add-vport` 命令的更多信息，请参见[“将 VPort 添加到弹性虚拟交换机” \[114\]](#)。

注 - 您无需始终将虚拟端口添加到弹性虚拟交换机。创建 VNIC 时，您只能指定 VNIC 必须连接到的弹性虚拟交换机的名称。在这种情况下，EVS 控制器将生成系统虚拟端口。这些虚拟端口遵循命名约定 `sys-vportname`，例如 `sys-vport0`。系统虚拟端口继承弹性虚拟交换机的属性。

5. (可选) 显示所配置的弹性虚拟交换机。

```
# evsadm
```

例 6-3 配置弹性虚拟交换机

以下示例说明了如何创建弹性虚拟交换机 ORA，并将 IPnet `ora_ipnet` 和 VPort `vport0` 添加到该弹性虚拟交换机。

```
# evsadm create-ivs ORA
# evsadm add-ipnet -p subnet=192.168.10.0/24 ORA/ora_ipnet
# evsadm add-vport ORA/vport0
# evsadm
```

NAME	TENANT	STATUS	VNIC	IP	HOST
ORA	sys-global	idle	--	ora_ipnet	--
vport0	--	free	--	192.168.10.2/24	--

以下示例说明了如何创建租户为 `tenantA` 的弹性虚拟交换机 ORA，并将 IPnet `ora_ipnet` 和 VPort `vport0` 添加到该弹性虚拟交换机。

```
# evsadm create-ivs -T tenantA ORA
# evsadm add-ipnet -T tenantA -p subnet=192.168.10.0/24 ORA/ora_ipnet
# evsadm add-vport -T tenantA ORA/vport0
# evsadm
```

NAME	TENANT	STATUS	VNIC	IP	HOST
ORA	tenantA	idle	--	ora_ipnet	--
vport0	--	free	--	192.168.10.2/24	--

为弹性虚拟交换机创建 VNIC

现在使用 `dladm` 和 `zonecfg` 命令可以为弹性虚拟交换机创建 VNIC。

▼ 如何为弹性虚拟交换机创建 VNIC

开始之前 必须使用 `evsadm set-prop` 命令在 EVS 节点上设置 `controller` 属性。有关更多信息，请参见[如何配置 EVS 控制器 \[110\]](#)。

1. 成为具有“弹性虚拟交换机管理”权限配置文件的管理员或用户。
有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 为弹性虚拟交换机配置 VNIC。

```
# dladm create-vnic -t -c EVS-switch-name[/VPort-name] [-T tenant-name] VNIC-name
```

`-t` 指定 VNIC 为临时 VNIC。

`-c EVS-switch-name[/VPort-name]` 指定必须将 VNIC 连接到的弹性虚拟交换机的名称。如果指定了 VPort 的名称，则 VNIC 将连接到指定的 VPort。如果未指定 VPort 名称，则系统将自动生成一个 VPort 并将该 VPort 指定给 VNIC。VNIC 在连接到弹性虚拟交换机后，VNIC 将从指定的弹性虚拟交换机或 VPort 继承属性。

`-T tenant-name` 指定拥有弹性虚拟交换机的租户的名称。如果未指定租户，则系统采用缺省的 `sys-global` 租户。

`VNIC-name` VNIC 的名称。

3. (可选) 显示有关连接到弹性虚拟交换机的 VNIC 的信息。

```
# dladm show-vnic -c
```

`-c` 选项显示有关连接到弹性虚拟交换机的 VNIC 的信息。

例 6-4 为弹性虚拟交换机创建 VNIC

此示例说明了如何创建临时的 VNIC `vnic1` 并将该 VNIC 连接到弹性虚拟交换机 `ORA` 和 VPort `vport0`。

```
# dladm create-vnic -t -c ORA/vport0 vnic1
# dladm show-vnic -c
LINK      TENANT      EVS      VPORT      OVER      MACADDRESS      VIDS
vnic1     sys-global  ORA      vport0     evs-vxlan10000  2:8:20:b0:6e:63  0
```

为弹性虚拟交换机创建 VNIC `anet` 资源

可以使用增强的 `zonecfg` 命令为弹性虚拟交换机配置区域的 VNIC `anet` 资源。

在配置区域时，可以为 anet 资源设置以下属性：

- tenant – 指定租户的名称。如果在配置区域时未指定值，则系统将指定缺省值，即 sys-global 租户。
- vport – 指定 VPort 的名称。如果在配置区域时未指定值，则将自动为弹性虚拟交换机生成系统 VPort，并且 VPort 将继承弹性虚拟交换机的属性。
- evs – 指定必须将 anet 资源连接到的弹性虚拟交换机的名称。

数据中心内的 VPort 由租户名称、弹性虚拟交换机名称和 VPort 名称唯一地予以标识。有关更多信息，请参见《[创建和使用 Oracle Solaris 区域](#)》。

例 6-5 为弹性虚拟交换机创建 VNIC anet 资源

此示例说明了如何创建一个具有 VNIC anet 资源 evszone/net1 的区域，该资源连接到租户 tenantA 的 ORA 和 vport0。

```
# zonecfg -z evszone
Use 'create' to begin configuring a new zone
zonecfg:evszone> create
create: Using system default template 'SYSdefault'
zonecfg:evszone> set zonepath=/export/zones/evszone
zonecfg:evszone> set tenant=tenantA
zonecfg:evszone> add anet
zonecfg:evszone:net> set evs=ORA
zonecfg:evszone:net> set vport=vport0
zonecfg:evszone:net> end
zonecfg:evszone> exit
# zoneadm -z evszone install
# zoneadm -z evszone boot
# zlogin -C evszone
# dladm show-vnic -c
LINK          TENANT  EVS  VPORT  OVER  MACADDRESS      VIDS
evszone/net1  tenantA  ORA  vport0 net2  2:8:20:89:a1:97  200
```

在引导 evszone 时，VNIC anet evszone/net1 将与 VPort ORA/vport0 的 MAC 地址、IP 地址和 SLA 属性关联。有关为弹性虚拟交换机配置区域的 VNIC anet 资源的更多信息，请参见“[使用案例：配置弹性虚拟交换机](#)” [130]。

管理弹性虚拟交换机、IPnet 和 VPort

本节介绍了如何管理弹性虚拟交换机、IPnet 和 VPort。有关如何配置弹性虚拟交换机、IPnet 和 VPort 的更多信息，请参见“[配置弹性虚拟交换机](#)” [112]。

管理弹性虚拟交换机

本节介绍了如何为弹性虚拟交换机执行以下任务：

- 显示有关弹性虚拟交换机的信息
- 设置弹性虚拟交换机的属性
- 显示弹性虚拟交换机的属性

显示弹性虚拟交换机信息

可以使用 `evsadm show-evs` 命令显示弹性虚拟交换机信息。命令语法为：

```
# evsadm show-evs [-f {fname=value[,...]}[,...]] [-L] [[-c] -o field[,...]] [EVS-switch-name]
```

`-f {fname=value[,...]}[,...]` 用来对输出进行过滤（行选择）的以逗号分隔的名称/值对。如果指定了多个过滤器，则所显示的输出是对过滤器执行 AND 操作的结果。如果过滤器值是多值的，则所显示的输出是对过滤器值执行 OR 操作的结果。支持的过滤器包括：

- `tenant`
- `evs`
- `host`
- `ipnet`
- `vport`

`-L` 显示与弹性虚拟交换机关联的 VLAN ID 或 VXLAN 网段 ID。

`-o field[,...]` 指定要显示的输出字段的逗号分隔列表，不区分大小写。可以指定以下字段，它们在输出中显示为列：

<code>all</code>	显示所有输出字段。
<code>EVS</code>	弹性虚拟交换机的名称。
<code>TENANT</code>	拥有弹性虚拟交换机的租户的名称。
<code>STATUS</code>	弹性虚拟交换机的状态，是空闲还是繁忙。如果弹性虚拟交换机至少有一个 VPort 连接有 VNIC，则它处于繁忙状态。
<code>NVPORTS</code>	与弹性虚拟交换机关联的虚拟端口的数量。
<code>IPNETS</code>	与 EVS 关联的 IP 网络的列表。目前，只有一个 IP 网络可以与弹性虚拟交换机关联。

HOST 弹性虚拟交换机在多台服务器之间跨越的主机的列表。

例 6-6 显示弹性虚拟交换机信息

下面的示例显示了弹性虚拟交换机 ORA 的信息。

```
# evsadm show-ivs ORA
EVS      TENANT      STATUS NVPORTS IPNETS      HOST
ORA      sys-global   busy   1         ora_ipnet   s11-client
```

下面的示例显示了与弹性虚拟交换机 ORA 关联的 VLAN ID。

```
# evsadm show-ivs -L
EVS      TENANT      VID  VNI
ORA      tenantA     200  --
```

该输出显示了以下信息：

EVS	弹性虚拟交换机的名称
TENANT	拥有弹性虚拟交换机的租户的名称
VID	用于实现弹性虚拟交换机的 VLAN ID
VNI	用于实现弹性虚拟交换机的 VXLAN 网段 ID

设置弹性虚拟交换机的属性

可以使用 `evsadm set-ivsprop` 命令设置弹性虚拟交换机的属性。命令语法为：

```
# evsadm set-ivsprop [-T tenant-name] -p prop=value[,...] EVS-switch-name
```

`-p prop` 在指定的弹性虚拟交换机上设置属性的值。

EVS 支持下面的属性：

- `maxbw` – 为连接到指定弹性虚拟交换机的所有虚拟端口设置全双工带宽。带宽指定为一个带有容量级后缀（分别表示 Kbps、Mbps 和 Gbps 的 K、M 或 G）的整数。如果未指定单位，输入值将以 Mbps 为单位进行读取。缺省值是无带宽限制。
- `priority` – 为连接到指定弹性虚拟交换机的所有虚拟端口设置缺省优先级。可能的值为 `high`、`medium` 或 `low`。缺省值为 `medium`。此优先级不会反映在线路上的任何协议优先级字段中，

而是用于系统内的包处理调度。高优先级 VPort 可能延迟更短，具体取决于可用的系统资源。

例 6-7 设置弹性虚拟交换机的属性

以下示例说明了如何设置弹性虚拟交换机 ORA 的属性。

```
# evsadm set-evsprop -p maxbw=200 ORA
# evsadm set-evsprop -p priority=high ORA
```

显示弹性虚拟交换机的属性

可以使用 `evsadm show-evsprop` 命令显示弹性虚拟交换机的属性。命令语法为：

```
# evsadm show-evsprop [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
[-p prop[,...]] [EVS-switch-name]
```

`-f {fname=value[,...]} [,...]` 用来对输出进行过滤（行选择）的以逗号分隔的名称/值对。如果指定了多个过滤器，则所显示的输出是对过滤器执行 AND 操作的结果。如果过滤器值是多值的，则所显示的输出是对过滤器值执行 OR 操作的结果。支持的过滤器包括：

- `tenant` – 按租户名称对弹性虚拟交换机属性进行过滤
- `evs` – 按弹性虚拟交换机名称对弹性虚拟交换机属性进行过滤
- `host` – 按主机名称对弹性虚拟交换机属性进行过滤

例 6-8 “显示弹性虚拟交换机的属性” 显示了基于过滤器值的输出。

`-o field[,...]` 指定要显示的输出字段的逗号分隔列表，不区分大小写。可以指定以下字段，它们在输出中显示为列：

<code>all</code>	显示所有输出字段。
<code>EVS</code>	弹性虚拟交换机的名称。
<code>TENANT</code>	拥有弹性虚拟交换机的租户的名称。
<code>PROPERTY</code>	弹性虚拟交换机属性的名称。
<code>PERM</code>	属性的读取或写入权限。显示的值为 <code>r</code> - 或 <code>rw</code> 。
<code>VALUE</code>	当前的属性值。如果未设置此值，此值将显示为 <code>--</code> 。如果此值未知，则显示为 <code>?</code> 。

DEFAULT	属性的缺省值。如果属性没有缺省值，则显示 --。
POSSIBLE	属性的可能值的逗号分隔列表。如果可能的值未知或无限制，则显示 --。

例 6-8 显示弹性虚拟交换机的属性

下面的示例显示了为弹性虚拟交换机 ORA 配置的属性。

```
# evsadm show-evsprop ORA
EVS      TENANT      PROPERTY  PERM VALUE  DEFAULT  POSSIBLE
ORA      sys-global  maxbw    rw  200      --      --
ORA      sys-global  priority rw  high    medium   low,medium,high
ORA      sys-global  tenant   r-  --      --      --
```

下面的示例显示了弹性虚拟交换机 HR 和 ORA 的输出。在此示例中，指定了 `evs` 过滤器来获取弹性虚拟交换机 HR 和 ORA 的输出。

```
# evsadm show-evsprop -f evs=HR,ORA
EVS      TENANT      PROPERTY  PERM VALUE  DEFAULT  POSSIBLE
HR       tenantA     maxbw    rw  300      --      --
HR       tenantA     priority rw  --      medium   low,medium,high
HR       tenantA     tenant   r-  --      --      --
ORA      sys-global  maxbw    rw  --      --      --
ORA      sys-global  priority rw  --      medium   low,medium,high
ORA      sys-global  tenant   r-  --      --      --
```

管理 IPnet 配置

本节说明了如何在为弹性虚拟交换机添加 IPnet 后对 IPnet 执行以下任务：

- 删除为弹性虚拟交换机配置的 IPnet
- 显示有关 IPnet 的信息

删除 IPnet

可以使用 `evsadm remove-ipnet` 命令删除为弹性虚拟交换机配置的 IPnet。命令语法为：

```
# evsadm remove-ipnet [-T tenant-name] EVS-switch-name/IPnet-name
```

此命令将从指定的弹性虚拟交换机中删除指定的 IPnet。如果其中任一 VPort 正在使用中，则无法删除 IPnet。如果 VPort 有连接到它的 VNIC，则说明它在使用。

例 6-9 删除为弹性虚拟交换机配置的 IPnet

此示例说明了如何从弹性虚拟交换机 ORA 中删除 IPnet ora_ipnet。

```
# evsadm remove-ipnet ORA/ora_ipnet
```

显示 IPnet

可以使用 `evsadm show-ipnet` 命令显示有关由 EVS 控制器管理的 IPnet 以及指定的 IPnet 的信息。命令语法为：

```
# evsadm show-ipnet [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] [IPnet-name]
```

`-f {fname=value[,...]}[,...]` 用来对输出进行过滤（行选择）的以逗号分隔的名称/值对。如果指定了多个过滤器，则所显示的输出是对过滤器执行 AND 操作的结果。如果过滤器值是多值的，则所显示的输出是对过滤器值执行 OR 操作的结果。支持的过滤器为 `tenant`、`evs`、`ipnet` 和 `host`。

`-o field[,...]` 指定要显示的输出字段的逗号分隔列表，不区分大小写。可以指定以下字段，它们在输出中显示为列：

<code>all</code>	显示所有输出字段。
<code>NAME</code>	IPnet 的名称及其关联的弹性虚拟交换机的名称。
<code>IPNET</code>	IPnet 的名称。
<code>EVS</code>	弹性虚拟交换机的名称。
<code>TENANT</code>	拥有弹性虚拟交换机的租户的名称。
<code>SUBNET</code>	表示此 IPnet 的子网（IPv4 或 IPv6）。
<code>START</code>	IP 地址范围的起始地址。
<code>END</code>	IP 地址范围的结束地址。
<code>DEFROUTER</code>	给定 IPnet 的缺省路由器的 IP 地址。
<code>AVAILRANGE</code>	可以分配给 VPort 的可用 IP 地址的逗号分隔列表。

例 6-10 显示弹性虚拟交换机的 IPnet

此示例显示了为弹性虚拟交换机 ORA 配置的 IPnet。

```
# evsadm show-ipnet
NAME          TENANT      SUBNET          DEFROUTER      AVAILRANGE
ORA/ora_ipnet sys-global 192.168.10.0/24 192.168.10.1 192.168.10.3-192.168.10.254
```

管理 VPort 配置

本节说明了如何对 VPort 执行以下任务：

- 设置 VPort 的属性
- 显示与 VPort 关联的属性
- 显示有关 VPort 的信息
- 重置 VPort
- 删除 VPort

设置 VPort 的属性

可以使用 `evsadm set-vportprop` 命令设置 VPort 的属性。命令语法为：

```
# evsadm set-vportprop [-T tenant-name] -p prop=value[,...] EVS-switch-name/VPort-name
```

`-T tenant-name` 指定租户的名称。

`-p prop=value[,...]` 为指定的 VPort 指定属性值。如果 VPort 有连接到它的 VNIC，则设置该 VPort 上的属性将导致 VNIC 的属性更改。有关 VPort 属性的信息，请参见表 5-1 “VPort 属性”。

注 - 无法更改系统 VPort 的属性。有关系统 VPort 的更多信息，请参见[如何配置弹性虚拟交换机 \[114\]](#)。

`EVS-switch-name/VPort-name` 指定要为其设置属性的弹性虚拟交换机或 VPort 的名称。

注 - 在创建 VPort 后，无法修改 `ipaddr`、`macaddr`、`evs` 和 `tenant` 属性。

例 6-11 设置 VPort 的属性

此示例说明了如何将 `HR/vport0` 的最大带宽属性设置为 1G。

```
# evsadm set-vportprop -p maxbw=1G HR/vport0
```

显示 VPort 的属性

可以使用 `evsadm show-vportprop` 命令显示 VPort 的属性。命令语法为：

```
# evsadm show-vportprop [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
[-p prop[,...]] [[EVS-switch-name]/[VPort-name]]
```

此命令显示所有 VPort 或指定 VPort 的一个或多个属性的当前值。如果未指定 VPort 属性，则将显示所有可用 VPort 属性。有关 VPort 属性的信息，请参见表 5-1 “VPort 属性”。

`[-f {fname=value[,...]}[,...]]` 用来对输出进行过滤（行选择）的以逗号分隔的名称/值对。如果指定了多个过滤器，则所显示的输出是对过滤器执行 AND 操作的结果。如果过滤器值是多值的，则所显示的输出是对过滤器值执行 OR 操作的结果。支持的过滤器包括：

- tenant – 按租户名称对 VPort 属性进行过滤
- EVS – 按弹性虚拟交换机名称对 VPort 属性进行过滤
- vport – 按 VPort 名称对 VPort 属性进行过滤
- host – 按主机名称对 VPort 属性进行过滤

`-o field[,...]` 指定要显示的输出字段的逗号分隔列表，不区分大小写。可以指定以下字段，它们在输出中显示为列：

all	显示所有输出字段。
NAME	VPort 的名称及其关联的弹性虚拟交换机的名称，格式为 <i>EVS-switch-name/VPort-name</i> 。
TENANT	拥有弹性虚拟交换机的租户的名称。
PROPERTY	VPort 属性的名称。
PERM	属性的读取或写入权限。显示的值为 r- 或 rw。
VALUE	当前的属性值。如果未设置此值，此值将显示为 --。在未知情况下，此值将显示为 ?。
DEFAULT	属性的缺省值。如果属性没有缺省值，则显示 --。

POSSIBLE

属性的可能值的逗号分隔列表。如果这些值在某个数字范围中，则可以显示简写形式 min-max。如果可能的值未知或无限制，则显示--。

例 6-12 显示 VPort 属性

此示例显示了 VPort vport0 的 VPort 属性。

```
# evsadm show-vportprop ORA/vport0
NAME          TENANT      PROPERTY  PERM VALUE          DEFAULT  POSSIBLE
ORA/vport0    sys-global  cos       rw  --                0        0-7
ORA/vport0    sys-global  maxbw     rw  --                --       --
ORA/vport0    sys-global  priority  rw  --                medium   low,medium,high
ORA/vport0    sys-global  ipaddr    r-  192.168.10.2/24  --       --
ORA/vport0    sys-global  macaddr   r-  2:8:20:b0:6e:63 --       --
ORA/vport0    sys-global  evs       r-  ORA              --       --
ORA/vport0    sys-global  tenant    r-  sys-global       --       --
```

显示 VPort

可以使用 `evsadm show-vport` 命令显示 VPort。命令语法为：

```
# evsadm show-vport [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
[[EVS-switch-name/]VPort-name]]
```

`-f {fname=value[,...]} [,...]` 用来对输出进行过滤（行选择）的以逗号分隔的名称/值对。如果指定了多个过滤器，则所显示的输出是对过滤器执行 AND 操作的结果。如果过滤器值是多值的，则所显示的输出是对过滤器值执行 OR 操作的结果。支持的过滤器包括：

- tenant – 按租户名称对 VPort 列表进行过滤
- EVS – 按弹性虚拟交换机名称对 VPort 列表进行过滤
- vport – 按 VPort 名称对 VPort 列表进行过滤
- host – 按主机名称对 VPort 列表进行过滤

`-o field[,...]` 指定要显示的输出字段的逗号分隔列表，不区分大小写。可以指定以下字段，它们在输出中显示为列：

all 显示所有输出字段。

NAME VPort 的名称及其关联的弹性虚拟交换机的名称，格式为 *EVS-switch-name/VPort-name*。

TENANT 拥有弹性虚拟交换机的租户的名称。

STATUS	显示 VPort 是正在使用中还是空闲。如果 VPort 与 VNIC 关联，则表示 VPort 正在使用中。否则，VPort 处于空闲状态。
VNIC	与 VPort 关联的 VNIC 的名称。
HOST	具有与 VPort 关联的 VNIC 的主机的名称。

例 6-13 显示 VPort 信息

此示例显示了有关 VPort vport0 的信息。

```
# evsadm show-vport
NAME          TENANT      STATUS VNIC      HOST
ORA/vport0    sys-global  used   vnic1     s11-client
```

重置 VPort

删除与 VPort 关联的 VNIC 时，VPort 的状态为 free。在下列情况下，即使删除了与 VPort 关联的 VNIC，VPort 也可以处于 used 状态：

- 删除 EVS 节点中的 VNIC 时，EVS 节点无法连接 EVS 控制器。
- 在重新引导 EVS 节点之前，与 VPort 关联的 VNIC 未删除。

要将 VPort 的状态重置为 free，请使用 evsadm reset-vport 命令。命令语法为：

```
# evsadm reset-vport [-T tenant-name] EVS-switch-name/VPort-name
```

删除 VPort

如果某个 VNIC 与该 VPort 关联，则无法删除该 VPort。因此，您必须首先使用 evsadm show-vport 命令检查 VNIC 是否与要删除的 VPort 关联。可以使用 evsadm remove-vport 命令从弹性虚拟交换机中删除 VPort。命令语法为：

```
# evsadm remove-vport [-T tenant-name] EVS-switch-name/VPort-name
```

此命令删除指定的 VPort。当删除某个 VPort 时，将释放与该 VPort 关联的 IP 地址和 MAC 地址。

例 6-14 删除 VPort

此示例说明了如何删除为弹性虚拟交换机 ORA 配置的 VPort vport0。

```
# evsadm remove-vport -T tenantA ORA/vport0
```

删除弹性虚拟交换机

本节介绍了如何删除弹性虚拟交换机。仅当弹性虚拟交换机的所有 VPort 均处于空闲状态时，才能删除弹性虚拟交换机。因此，VPort 必须未与 VNIC 关联。

▼ 如何删除弹性虚拟交换机

1. 成为具有“弹性虚拟交换机管理”权限配置文件的管理员或用户。
有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。
2. 检查弹性虚拟交换机是否在使用 VPort。

```
# evsadm show-evs
```

VPort 正在使用时不能删除弹性虚拟交换机。如果有 VNIC 连接到 VPort，则表示 VPort 正在使用中。evsadm show-evs 命令输出中的 STATUS 字段显示弹性虚拟交换机是繁忙还是空闲。

如果 VPort 正在使用中，则需要删除与 VPort 关联的 VNIC，如下所示：

```
# dladm delete-vnic VNIC
```

3. 删除弹性虚拟交换机。

```
# evsadm delete-evs [-T tenant-name] EVS-switch-name
```

此命令将删除指定的弹性虚拟交换机以及与该弹性虚拟交换机关联的所有 VPort 和 IPnet。

例 6-15 删除弹性虚拟交换机

以下示例说明了如何删除弹性虚拟交换机 ORA。

```
# evsadm show-evs
EVS      TENANT      STATUS NVPORTS IPNETS      HOST
ORA      sys-global  idle   0         ora_ipnet   --
# evsadm delete-evs ORA
# evsadm show-evs ORA
evsadm: failed to show EVS(s): evs not found
```

以下示例说明了如何删除处于繁忙状态的弹性虚拟交换机 EVS1。

```
# evsadm show-evs EVS1
EVS          TENANT          STATUS NVPORTS IPNETS      HOST
EVS1         sys-global    busy   1          evs1_ipnet s11-server
# evsadm show-vport EVS1/vport1
NAME          TENANT          STATUS VNIC          HOST
EVS1/vport1  sys-global    used   vnic1         s11-server
# dladm delete-vnic vnic1
# evsadm show-evs EVS1
EVS          TENANT          STATUS NVPORTS IPNETS      HOST
EVS1         sys-global    idle   1          evs1_ipnet --
# evsadm delete-evs EVS1
# evsadm show-evs EVS1
evsadm: failed to show EVS(s): evs not found
```

监视弹性虚拟交换机

可以监视弹性虚拟交换机的虚拟端口的网络通信统计数据以获取以下信息：

- VM 发送和接收的网络通信量，其提供有关 VM 上的工作负载的信息。
- 丢弃的传入 (idrops) 和传出 (odrops) 包的数量。这些值提供有关故障网络的信息。
- 由计算节点上的所有 VM 发送和接收的网络通信量，可帮助您执行容量规则。

可以使用 `evsstat` 命令监视弹性虚拟交换机。`evsstat` 命令报告弹性虚拟交换机的每个 VPort 的运行时统计数据。它还报告与 VPort 关联的 VNIC 的统计信息。有关 EVS 和虚拟端口的更多信息，请参见 [evsadm\(1M\)](#) 手册页。

`evsstat` 命令是一个远程管理守护进程 (Remote Administration Daemon, RAD) 客户机，它与远程 EVS 控制器通信以便执行所有 `evsstat` 子命令。在使用 `evsstat` 命令之前，必须使用 `evsadm set-prop` 命令指定 EVS 控制器的可解析的主机名或 IP 地址。命令语法为：

```
# evsadm set-prop -p controller=ssh://[username@]hostname-or-IP-address
```

此外，您还必须使用在运行 `evsstat` 命令的主机与 EVS 控制器之间预先共享的公钥来设置 SSH 验证。您需要采用具有预先共享的公钥的 SSH 验证，使 `evsstat` 命令能够以非交互模式安全地与 EVS 控制器通信。有关更多信息，请参见“[设置 SSH 验证](#)” [106]。

`evsstat` 的命令语法为：

```
# evsstat [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] [-u R|K|M|G|T|P] \
[EVS-switch-name[/VPort-name]] [interval] [count]
```

EVS-switch-name 指定要监视其统计数据的弹性虚拟交换机的名称。如果未指定弹性虚拟交换机的名称，则将显示所有弹性虚拟交换机的统计数据。

<i>VPort-name</i>	指定要监视其统计数据的 VPort 的名称。只会显示连接到指定 VPort 的 VNIC 的统计数据。必须指定弹性虚拟交换机的名称，然后指定 VPort 的名称。
-f {fname=val[,...]} [,...]	用来对输出进行过滤（行选择）的以逗号分隔的名称/值对。如果指定了多个过滤器，则所显示的输出是对过滤器执行 AND 操作的结果。如果过滤器值是多值的，则所显示的输出是对过滤器值执行 OR 操作的结果。支持的过滤器为 tenant、evs 和 host。
-o <i>field</i> [,...]	<p>指定要显示的输出字段的逗号分隔列表，不区分大小写。可以指定以下字段，它们在输出中显示为列：</p> <ul style="list-style-type: none"> ▪ vport ▪ evs ▪ tenant ▪ vnic ▪ host ▪ ipkts ▪ rbytes ▪ opkts ▪ idrops ▪ odrops
-u R K M G T P	<p>指定显示统计数据时所使用的单位。如果未指定，则将根据情况使用不同单位显示统计数据，采用 xy.zU 格式，其中 x、y 和 z 是数字，U 是相应的单位。支持的单位有：</p> <ul style="list-style-type: none"> ▪ R – 原始计数 ▪ K – KB ▪ M – MB ▪ G – GB ▪ T – TB ▪ P – PB
<i>interval</i>	指定您希望刷新网络统计数据的时间（秒）。
<i>count</i>	指定统计数据的刷新次数。必须指定时间间隔，然后指定计数。

例 6-16 监视弹性虚拟交换机

下面的示例显示了所有弹性虚拟交换机的统计数据。

```
# evsstat
VPORT      EVS      TENANT    IPKTS    RBYTES    OPKTS    OBYTES
```

```

sys-vport0  ORA      sys-global  101.88K  32.86M   40.16K  4.37M
sys-vport2  ORA      sys-global  4.50M    6.78G    1.38M   90.90M
sys-vport0  HR       sys-global  132.89K  12.25M   236     15.82K
sys-vport1  HR       sys-global  144.47K  13.32M   247     16.29K

```

下面的示例显示了指定弹性虚拟交换机 `evs0` 的统计数据。

```

# evsstat ORA
VPORT      EVS      TENANT      IPKTS      RBYTES      OPKTS      OBYTES
sys-vport0  ORA      sys-global  101.88K    32.86M      40.16K     4.37M
sys-vport2  ORA      sys-global  4.50M      6.78G       1.38M      90.90M

```

下面的示例显示了指定 VPort `evs0/sys-vport2` 的统计数据。

```

# evsstat ORA/sys-vport2
VPORT      EVS      TENANT      IPKTS      RBYTES      OPKTS      OBYTES
sys-vport2  ORA      sys-global  4.50M      6.78G       1.38M      90.90M

```

以下示例显示了 VPort 的统计数据，其时间间隔值为 1 秒，计数值为 3。在 1 秒钟的时间间隔内将刷新三次这些统计数据。

```

# evsstat ORA/sys-vport2 1 3
VPORT      EVS      TENANT      IPKTS      RBYTES      OPKTS      OBYTES
sys-vport2  ORA      sys-global  4.50M      6.78G       1.38M      90.90M
sys-vport2  ORA      sys-global  4.50M      6.78G       1.38M      90.90M
sys-vport2  ORA      sys-global  4.50M      6.78G       1.38M      90.90M

```

以下示例显示了指定输出字段的统计数据。

```

# evsstat -o vport,evs,vnic,host,ipkts,opkts
VPORT      EVS      VNIC      HOST      IPKTS      OPKTS
sys-vport0  ORA      vnic0     host1     101.88K    40.16K
sys-vport2  ORA      vnic0     host2     4.50M      1.38M
sys-vport0  HR       vnic1     host1     132.89K    236
sys-vport1  HR       vnic1     host2     144.47K    247

```

弹性虚拟交换机的使用案例示例

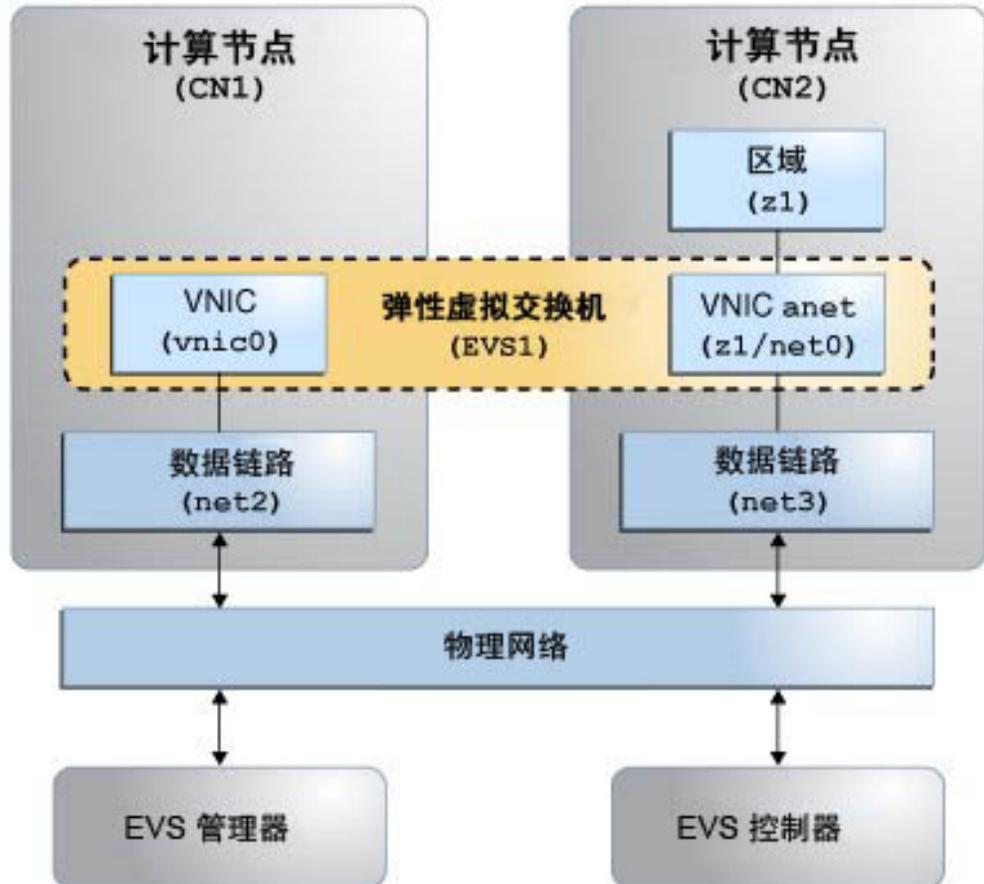
本节提供了使用案例示例来说明如何配置弹性虚拟交换机。

使用案例：配置弹性虚拟交换机

目标 – 此使用案例说明了如何设置一个跨越两个计算节点的弹性虚拟交换机 (EVS1)。

在此使用案例中，您将 CN1 上的 VNIC `vnic0` 和区域 `z1` 的 VNIC `anet` 连接到弹性虚拟交换机 `EVS1`，使它们属于同一个 L2 网段并且可以在 VLAN 上彼此相互通信。下图显示了跨越两个计算节点的弹性虚拟交换机 (EVS1)。

图 6-2 弹性虚拟交换机配置



此图显示了一个具有四个节点且包含以下组件的网络：

- 两个计算节点 (CN1 和 CN2)
- CN2 上具有 VNIC anet 资源的区域 z1 (z1/net0)
- CN1 上的 VNIC vnic0
- 一个充当 EVS 控制器的节点 (evs-controller.example.com)
- 一个充当您需要在其上运行 evsadm 命令的 EVS 管理器的节点 (MANAGER)
- 一个用于实现弹性虚拟交换机 EVS1 的 VLAN
- uplink-port，指定用于 VLAN 的数据链路

注 - 这四个节点可以都位于一台计算机上。EVS 控制器和 EVS 管理器可以位于同一台计算机上。

规划弹性虚拟交换机设置

1. 安装必需的 EVS 软件包。

有关所需软件包的信息，请参见[“使用 EVS 时必需的软件包” \[98\]](#)。

注 - `evsuser` 是在安装 `pkg:/service/network/evs` 软件包时创建的特定用户。用户 `evsuser` 指定有“弹性虚拟交换机管理”权限配置文件。此配置文件提供用于执行 EVS 操作的所有授权和特权。

2. 在 EVS 设置中，使用在以下组件之间为 `evsuser` 预先共享的公钥来设置 SSH 验证：

- EVS 管理器和 EVS 控制器
- 每个 EVS 节点和 EVS 控制器
- EVS 控制器和每个 EVS 节点

有关更多信息，请参见[“设置 SSH 验证” \[106\]](#)。

注 - 该用例假定 `controller` 属性设置为 EVS 节点、EVS 管理器和 EVS 控制器上的 `ssh://evsuser@evs-controller.example.com`。

3. 配置 EVS 控制器。

- a. 指定一个计算节点作为网络中的 EVS 控制器，然后在每个计算节点上设置 EVS 控制器，以便这些计算节点可以与 EVS 控制器通信。请注意，您可以从与 EVS 控制器通信的任何计算节点设置控制器属性。有关更多信息，请参见[“配置弹性虚拟交换机” \[112\]](#)。
- b. 指定属性 `l2-type`、`vlan-range` 和 `uplink-port`。否则，您无法创建弹性虚拟交换机。

4. 创建弹性虚拟交换机。您必须关联一个 IPnet 并将 VPort 添加到弹性虚拟交换机中。

5. 在 `cn1` 上创建一个临时 VNIC，并将该 VNIC 连接到弹性虚拟交换机的 VPort。

6. 在区域 `z1` 上创建一个 VNIC `anet` 资源并将其连接到弹性虚拟交换机。

EVS 管理器操作

1. 设置 EVS 控制器。

```
MANAGER# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 设置 EVS 控制器属性。

- a. 设置必须用于弹性虚拟交换机的 L2 拓扑类型。

```
MANAGER# evsadm set-controlprop -p l2-type=vlan
```

- b. 设置 VLAN 范围。

```
MANAGER# evsadm set-controlprop -p vlan-range=200-300
```

- c. 指定用于 VLAN 的数据链路 (uplink-port)。

```
MANAGER# evsadm set-controlprop -p uplink-port=net2
```

```
MANAGER# evsadm set-controlprop -h CN2 -p uplink-port=net3
```

注 - 您可以从数据中心的任何节点配置 EVS 控制器，只要您可以连接到 EVS 控制器并且具有必需的授权即可。有关更多信息，请参见[“使用 EVS 时的安全要求” \[99\]](#)。

3. 验证控制器属性。

```
MANAGER# evsadm show-controlprop -p l2-type,vlan-range,uplink-port
```

NAME	VALUE	DEFAULT	HOST
l2-type	vlan	vlan	--
vlan-range	200-300	--	--
uplink-port	net2	--	--
uplink-port	net3	--	CN2

4. 创建名为 EVS1 的弹性虚拟交换机。

```
MANAGER# evsadm create-evs EVS1
```

5. 将 IPnet EVS1_ipnet 添加到 EVS1。

```
MANAGER# evsadm add-ipnet -p subnet=192.168.100.0/24 EVS1/EVS1_ipnet
```

6. 将 VPort vport0 添加到 EVS1。

```
MANAGER# evsadm add-vport EVS1/vport0
```

您无需始终将虚拟端口添加到弹性虚拟交换机。创建 VNIC 时，您只能指定 VNIC 必须连接到的弹性虚拟交换机的名称。在这种情况下，EVS 控制器将生成系统虚拟端口。这些虚拟端口遵循命名约定 `sys-vportname`，例如 `sys-vport0`。系统虚拟端口继承弹性虚拟交换机的属性。

7. 验证所创建的弹性虚拟交换机。

```
MANAGER# evsadm
```

NAME	TENANT	STATUS	VNIC	IP	HOST
EVS1	sys-global	--	--	EVS1_ipnet	--
vport0	--	free	--	192.168.100.2/24	--

注 - 因为未指定租户名称，所以弹性虚拟交换机 EVS1 使用缺省的租户名称 `sys-global`。在创建弹性虚拟交换机时，可以使用 `-T` 选项指定租户名称。有关更多信息，请参见[如何配置弹性虚拟交换机 \[114\]](#)。

8. 检查与 EVS1/vport0 关联的 MAC 地址和 IP 地址。

```
MANAGER# evsadm show-vportprop -p macaddr,ipaddr EVS1/vport0
```

NAME	TENANT	PROPERTY	PERM	VALUE	DEFAULT	POSSIBLE
EVS1/vport0	sys-global	ipaddr	r-	192.168.100.2/24	--	--
EVS1/vport0	sys-global	macaddr	r-	2:8:20:3c:78:bd	--	--

连接到 vport0 的 VNIC 将继承该 IP 地址和 MAC 地址。为 vport0 指定的 IP 地址是 IPnet EVS1_ipnet 中下一个可用的 IP 地址，而 MAC 地址则是为 vport0 随机生成的。

9. 检查与弹性虚拟交换机 EVS1 关联的 VLAN ID。

```
MANAGER# evsadm show-evs -L
```

EVS	TENANT	VID	VNI
EVS1	sys-global	200	--

计算节点 CN1 操作

1. 指定 EVS 控制器。

```
CN1# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 创建一个临时 VNIC vnic0 并将其连接到 EVS1/vport0。

```
CN1# dladm create-vnic -t -c EVS1/vport0 vnic0
```

3. 验证所创建的 VNIC。

```
CN1# dladm show-vnic -c
```

LINK	TENANT	EVS	VPORT	OVER	MACADDRESS	VIDS
vnic0	sys-global	EVS1	vport0	net2	2:8:20:3c:78:bd	200

vnic0 的 MAC 地址映射到 VPort 的 MAC 地址。

4. 检查 vnic0 允许的 IP 地址。

```
CN1# dladm show-linkprop -p allowed-ips vnic0
```

LINK	PROPERTY	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
vnic0	allowed-ips	192.168.100.2	192.168.100.2	--	--

allowed-ips 属性设置为与 VPort 关联的 IP 地址。在此设置下，您无法在 vnic0 上创建除 192.168.100.2 之外的任何其他 IP 地址。

5. 为 vnic0 创建 IP 接口，并将 192.168.100.2 指定为 IP 地址。

```
# ipadm create-ip -t vnic0
# ipadm create-addr -t -a 192.168.100.2 vnic0
```

计算节点 CN2 操作

1. 指定 EVS 控制器。

```
CN2# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 为区域 z1 配置 VNIC anet 资源并将其连接到弹性虚拟交换机。

```
CN2# zonecfg -z z1
Use 'create' to begin configuring a new zone
zonecfg:z1> create
create: Using system default template 'SYSdefault'
zonecfg:z1> set zonepath=/export/zones/z1
zonecfg:z1> select anet linkname=net0
zonecfg:z1:anet> set evs=EVS1
zonecfg:z1:anet> end
zonecfg:z1> commit
zonecfg:z1> exit
```

3. 安装并引导区域 z1。

```
CN2# zoneadm -z z1 install
CN2# zoneadm -z z1 boot
```

4. 登录到区域 z1 并完成区域配置。

```
CN2# zlogin -C z1
```

有关区域配置的更多信息，请参见《[创建和使用 Oracle Solaris 区域](#)》。

5. 验证所创建的 VNIC anet。

```
CN2# dladm show-vnic -c
LINK          TENANT      EVS   VPORT      OVER  MACADDRESS      VIDS
z1/net0       sys-global EVS1  sys-vport0 net2   2:8:20:1a:c1:e4  200
```

由于您在创建 VNIC anet 资源时未指定 VPort，因此 EVS 控制器将为 VNIC anet 资源创建一个系统 VPort sys-vport0。

6. 显示与 VPort 相关的信息。

```
CN2# evsadm show-vport -o all
NAME          TENANT      STATUS VNIC      HOST MACADDR      IPADDR
EVS1/sys-vport0 sys-global used   z1/net0 CN2  2:8:20:1a:c1:e4  192.168.100.3/24
```

VNIC anet 资源已激活并为其指定了 VPort 的 IP 地址。

7. 验证 VNIC anet 资源 z1/net0 的 IP 地址。

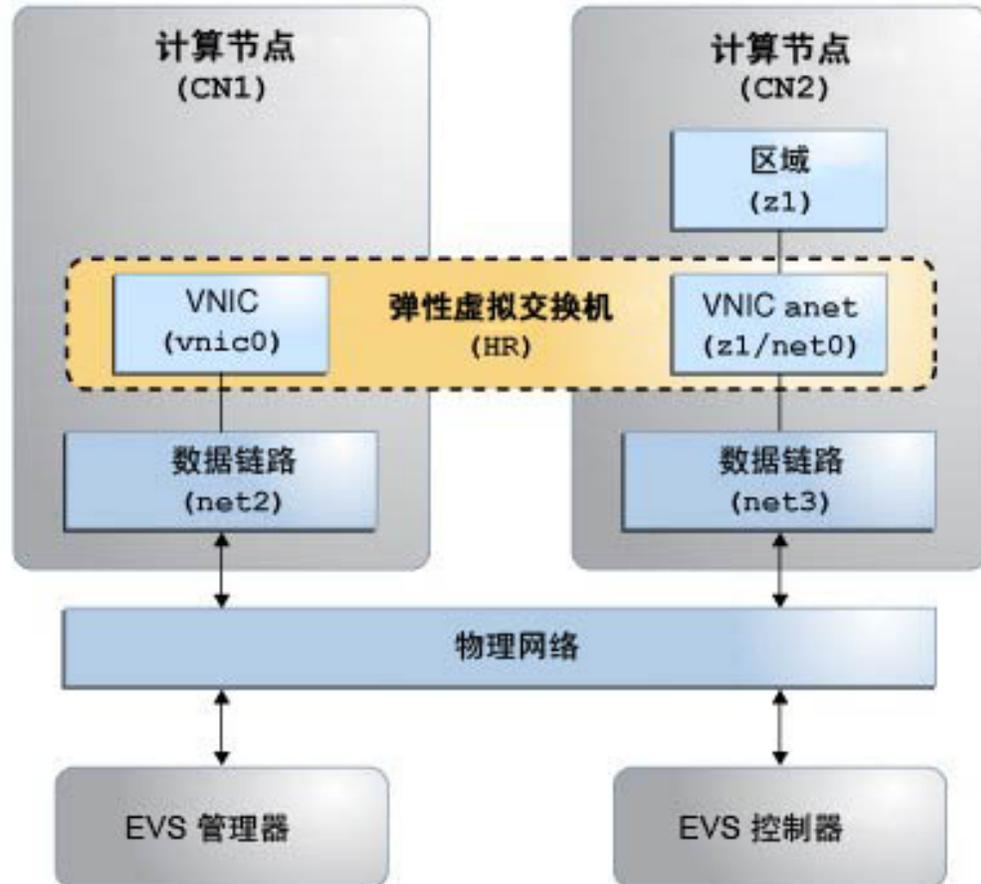
```
CN2# zlogin z1 ipadm
NAME          CLASS/TYPE  STATE  UNDER  ADDR
lo0           loopback   ok     --      --
  lo0/v4      static     ok     --      127.0.0.1/8
  lo0/v6      static     ok     --      ::1/128
net0          ip         ok     --      --
  net0/v4     inherited  ok     --      192.168.100.3/24
```

使用案例：为租户配置弹性虚拟交换机

目标 - 此使用案例说明了如何为租户设置一个跨越两个计算节点的弹性虚拟交换机 (HR)。

在此使用案例中，您将 CN1 上的 VNIC vnic0 和区域 z1 的 VNIC anet 连接到弹性虚拟交换机 HR，使它们属于同一 L2 网段并且可以在 VXLAN 上彼此相互通信。这些 VNIC 属于租户 tenantA。下图显示了 EVS 设置。

图 6-3 租户的弹性虚拟交换机配置



此图显示了一个具有四个节点且包含以下组件的网络：

- 两个计算节点 (CN1 和 CN2)
- CN2 上具有 VNIC anet 资源的区域 z1
- CN1 上的 VNIC vnic0
- 一个充当 EVS 控制器的节点 CONTROLLER
- 一个充当您需要在其上运行 `evsadm` 命令的 EVS 管理器的节点 MANAGER
- 一个用于实现弹性虚拟交换机 HR 的 VXLAN
- `uplink-port`，指定用于 VXLAN 的数据链路

规划弹性虚拟交换机设置

1. 安装必需的 EVS 软件包。有关所需软件包的信息，请参见[“使用 EVS 时必需的软件包” \[98\]](#)。

注 - evsuser 是在安装 pkg:/service/network/evs 软件包时创建的特定用户。用户 evsuser 指定有“弹性虚拟交换机管理”权限配置文件。此配置文件提供用于执行 EVS 操作的所有授权和特权。

2. 在 EVS 设置中，使用在以下组件之间为 evsuser 预先共享的公钥来设置 SSH 验证：
 - EVS 管理器和 EVS 控制器
 - 每个 EVS 节点和 EVS 控制器
 - EVS 控制器和每个 EVS 节点

有关更多信息，请参见[“设置 SSH 验证” \[106\]](#)。

注 - 该用例假定 controller 属性设置为每个 EVS 节点、EVS 管理器和 EVS 控制器上的 ssh://evsuser@evs-controller.example.com。

3. 配置 EVS 控制器并设置控制器属性。
 - a. 在所有计算节点上设置 EVS 控制器，然后设置控制器属性以指定如何跨计算节点实现弹性虚拟交换机。
 - b. 指定属性 l2-type、vxlan-range 和 uplink-port。否则，您无法创建弹性虚拟交换机。
4. 创建弹性虚拟交换机。您必须关联一个 IPnet 并将 VPort 添加到弹性虚拟交换机中。
5. 在 cn1 上创建一个临时 VNIC，并将该 VNIC 连接到弹性虚拟交换机的 VPort。
6. 在区域 z1 上创建 VNIC anet 并将 VNIC anet 资源连接到弹性虚拟交换机。

EVS 管理器操作

1. 设置 EVS 控制器。

```
MANAGER# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 设置 EVS 控制器属性。

- a. 设置必须用于弹性虚拟交换机的 L2 拓扑类型。此示例使用了 VXLAN。

```
MANAGER# evsadm set-controlprop -p l2-type=vxlan
```

- b. 设置 VXLAN 范围。

```
MANAGER# evsadm set-controlprop -p vxlan-range=200-300
```

- C. 指定用于 VXLAN 的数据链路 (uplink-port)。

```
MANAGER# evsadm set-controlprop -p uplink-port=net2
```

```
MANAGER# evsadm set-controlprop -h CN2 -p uplink-port=net3
```

注 - 您可以从数据中心的任何节点配置控制器，只要您可以连接到 EVS 控制器并且具有必需的授权即可。有关更多信息，请参见“使用 EVS 时的安全要求” [99]。

3. 验证 EVS 控制器属性。

```
MANAGER# evsadm show-controlprop -p l2-type,vxlan-range,uplink-port
```

NAME	VALUE	DEFAULT	HOST
l2-type	vxlan	vlan	--
vxlan-range	200-300	--	--
uplink-port	net2	--	--
uplink-port	net3	--	CN2

4. 为租户 tenantA 创建弹性虚拟交换机 HR。

```
MANAGER# evsadm create-evs -T tenantA HR
```

5. 将 IPnet hr_ipnet 添加到弹性虚拟交换机 HR。

```
MANAGER# evsadm add-ipnet -T tenantA -p subnet=192.168.100.0/24 HR/hr_ipnet
```

6. 将 VPort vport0 添加到弹性虚拟交换机 HR。

```
MANAGER# evsadm add-vport -T tenantA HR/vport0
```

7. 验证为租户 tenantA 创建的弹性虚拟交换机。

```
MANAGER# evsadm
```

NAME	TENANT	STATUS	VNIC	IP	HOST
HR	tenantA	--	--	hr_ipnet	--
vport0	--	free	--	192.168.100.2/24	--

8. 检查与 HR/vport0 关联的 MAC 地址和 IP 地址。

```
MANAGER# evsadm show-vportprop -p macaddr,ipaddr HR/vport0
```

NAME	TENANT	PROPERTY	PERM	VALUE	DEFAULT	POSSIBLE
HR/vport0	tenantA	ipaddr	r-	192.168.100.2/24	--	--
HR/vport0	tenantA	macaddr	r-	2:8:20:d8:da:10	--	--

9. 检查与弹性虚拟交换机 HR 关联的 VXLAN 网段 ID。

```
MANAGER# evsadm show-evs -L
```

EVS	TENANT	VID	VNI
HR	tenantA	--	200

计算节点 CN1 操作

1. 指定 EVS 控制器。

```
CN1# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 创建临时的 VNIC vnic0 并将其连接到弹性虚拟交换机 HR/vport0。

```
CN1# dladm create-vnic -t -T tenantA -c HR/vport0 vnic0
```

3. 验证所创建的 VNIC。

```
CN1# dladm show-vnic -c
LINK  TENANT  EVS  VPORT  OVER  MACADDRESS  VIDS
vnic0 tenantA HR  vport0 evs-vxlan200 2:8:20:d8:da:10 0
```

vnic0 的 MAC 地址映射到 VPort 的 MAC 地址。

4. 检查 vnic0 允许的 IP 地址。

```
CN1# dladm show-linkprop -p allowed-ips vnic0
LINK  PROPERTY  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
vnic0  allowed-ips 192.168.100.2 192.168.100.2 -- --
```

allowed-ips 属性设置为与 VPort 关联的 IP 地址。此输出意味着您无法在 vnic0 上创建除 192.168.100.2 之外的任何 IP 地址。

5. 为 vnic0 创建 IP 接口，并将 192.168.100.2 指定为 IP 地址。

```
# ipadm create-ip -t vnic0
# ipadm create-addr -t -a 192.168.100.2 vnic0
```

6. 检查自动生成的 VXLAN 数据链路。

```
CN1# dladm show-vxlan
LINK  ADDR  VNI  MGROUP
evs-vxlan200 0.0.0.0 200 224.0.0.1
```

计算节点 CN2 操作

1. 指定 EVS 控制器。

```
CN2# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 为区域 z1 配置 VNIC anet 并将其连接到弹性虚拟交换机。

```
CN2# zonecfg -z z1
Use 'create' to begin configuring a new zone
zonecfg:z1> create
create: Using system default template 'SYSdefault'
zonecfg:z1> set zonepath=/export/zones/z1
zonecfg:z1> set tenant=tenantA
```

```

zonecfg:z1> select anet linkname=net0
zonecfg:z1:anet> set evs=HR
zonecfg:z1:anet> end
zonecfg:z1> commit
zonecfg:z1> exit

```

3. 安装并引导区域 z1。

```

CN2# zoneadm -z z1 install
CN2# zoneadm -z z1 boot

```

4. 登录到区域 z1 并完成区域配置。

```

CN2# zlogin -C z1

```

有关区域配置的更多信息，请参见《[创建和使用 Oracle Solaris 区域](#)》。

5. 验证所创建的 VNIC anet 资源。

```

CN2# dladm show-vnic -c
LINK      TENANT  EVS   VPORT      OVER      MACADDRESS      VIDS
z1/net0   tenantA HR    sys-vport0 evs-vxlan200 2:8:20:1a:c1:e4 0

```

因为未指定 VPort，所以 EVS 控制器将为 VNIC anet 资源创建一个系统 VPort sys-vport0。

6. 显示与 VPort 相关的信息。

```

CN2# evsadm show-vport -o all
NAME          TENANT  STATUS VNIC      HOST MACADDR      IPADDR
HR/sys-vport0 tenantA used   z1/net0  CN2  2:8:20:1a:c1:e4 192.168.100.3/24

```

VNIC anet 资源已激活并为其指定了 VPort 的 IP 地址。

7. 验证 VNIC anet z1/net0 的 IP 地址。

```

CN2# zlogin z1 ipadm
NAME          CLASS/TYPE STATE      UNDER  ADDR
lo0           loopback  ok         --      --
  lo0/v4      static    ok         --      127.0.0.1/8
  lo0/v6      static    ok         --      ::1/128
net0          ip        ok         --      --
  net0/v4    inherited ok         --      192.168.100.3/24

```


管理网络资源

本章说明了如何使用数据链路属性和流来管理和分配网络资源。通过管理网络资源，可以实现 IP 服务质量 (quality of service, QoS)，从而提高虚拟网络和物理网络的性能。有关网络资源管理的介绍，请参见[“网络资源管理概述” \[18\]](#)。

本章包含以下主题：

- [“使用数据链路属性管理网络资源” \[143\]](#)
- [“管理 NIC 环” \[144\]](#)
- [“管理池和 CPU” \[152\]](#)
- [“使用流管理网络资源” \[157\]](#)
- [“使用案例：通过设置数据链路和流属性来管理网络资源” \[160\]](#)

使用数据链路属性管理网络资源

可以向数据链路分配网络资源以提高系统对包的处理效率。可以在创建数据链路时通过设置数据链路属性来分配网络资源。另外，也可以将数据链路属性设置为现有数据链路。可以通过使用 `dladm` 命令设置下面的数据链路属性，将网络资源分配给数据链路：

- `maxbw` – 指定可分配给数据链路的最大带宽。有关更多信息，请参见[“使用案例：通过设置数据链路和流属性来管理网络资源” \[160\]](#)。
- `rxrings` 和 `txrings` – 指定可分配给特定数据链路的 NIC 的接收环数和传送环数。有关更多信息，请参见[“管理 NIC 环” \[144\]](#)。
- `pool` – 指定 CPU 池的名称，该池包含可分配给数据链路以便高效管理网络进程的 CPU 集。有关更多信息，请参见[“管理池和 CPU” \[152\]](#)。
- `cpus` – 指定可分配给数据链路的 CPU 的名称。有关更多信息，请参见[“管理池和 CPU” \[152\]](#)。

有关在 Oracle Solaris 中管理网络资源的演示，请参见 [Managing Network Resources Using Oracle Solaris \(http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/ManagingNetworkResources/ManagingNetworkResources.htm\)](http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/ManagingNetworkResources/ManagingNetworkResources.htm) (使用 Oracle Solaris 管理网络资源)。

用于在数据链路中分配资源的命令

下面的命令用于在数据链路中分配网络资源：

- 要创建一个虚拟链路并同时向其分配资源，请使用以下命令语法：

```
# dladm create-vnic -l link -p prop=value[,...] VNIC
```

link 指链路的名称，可以是物理链路也可以是虚拟链路。

prop 指数据链路属性。有关可以为资源分配设置的各种数据链路属性的信息，请参见“[使用数据链路属性管理网络资源](#)” [143]。

- 要设置现有链路的属性，请使用以下命令语法：

```
# dladm set-linkprop -p prop=value[,...] link
```

有关更多信息，请参见 [dladm\(1M\)](#) 手册页。

管理 NIC 环

在 NIC 上，接收 (Rx) 环和传送 (Tx) 环是硬件资源，系统分别通过它们来接收和传送网络数据包。通过根据网络通信管理和分配环，可以提高系统对包的处理效率。例如，可以为要接收更多包的链路分配更多数量的接收 (Rx) 环。

在 MAC 客户机中分配环

MAC 客户机（例如，物理数据链路和 VNIC）是基于 NIC 配置的，用于实现系统与其他网络节点之间的通信。MAC 客户机可以是基于硬件的客户机或基于软件的客户机。

基于硬件的客户机

独占使用一个或多个 NIC 环的客户机称为基于硬件的客户机。您可以根据 NIC 支持的环分配来指定供基于硬件的客户机独占使用的环。

基于软件的客户机

不独占使用 NIC 环的客户机称为基于软件的客户机。它们与其他基于软件的现有客户机或主客户机共享环。基于软件的客户机所使用的环取决于在环分配中优先级高的基于硬件的客户机的数量。

在 VLAN 中分配环

VLAN 中的环分配因 VLAN 的创建方式而异。

您可以通过以下方式创建 VLAN：

- 通过使用 `dladm create-vlan` 命令：

```
# dladm create-vlan -l link -v vid VLAN
```

如果使用 `dladm create-vlan` 命令创建 VLAN，则它将与底层数据链路共享相同的 MAC 地址。因此，该 VLAN 还共享底层数据链路的 Rx 和 Tx 环。有关配置 VLAN 的更多信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的“配置 VLAN”。

- 通过使用 `dladm create-vnic` 命令：

```
# dladm create-vnic -l link -v vid VNIC
```

如果使用 `dladm create-vnic` 命令将 VLAN 创建为 VNIC，则其 MAC 地址与底层数据链路的不同。此类型 VLAN 的环分配独立于底层数据链路的分配。因此，可以为此 VLAN 指定它自己的专用环（假定 NIC 支持基于硬件的客户机）。有关如何为客户机指定环的更多信息，请参见“配置客户机并分配环” [147]。

用于配置环的命令

要配置数据链路的环，请使用下面的 `dladm` 子命令：

- `# dladm show-linkprop link`

显示数据链路属性的当前值，包括 Rx 和 Tx 环。有关示例，请参见例 7-1 “数据链路上的环使用和环分配”。

下表介绍了使用 `dladm show-linkprop` 命令显示的环属性。

环属性	权限	说明
<code>rxringsavail</code>	只读	指示可以分配给物理数据链路上基于硬件的客户机的 Rx 环数。
<code>rxhwclntavail</code>	只读	指示可以在物理数据链路上创建的基于硬件的 Rx 客户机数。
<code>rxrings</code>	读写	指示数据链路独占使用的 Rx 环数。可以将此属性设置为三个可能的值之一： <ul style="list-style-type: none"> ■ <code>hw</code> 指示您正在配置基于硬件的客户机。如果底层物理链路上的基于硬件的 Rx 客户机数 (<code>rxhwclntavail</code>) 大于零，则可以设置此值。

环属性	权限	说明
		<ul style="list-style-type: none"> ■ <i>number</i> 指示可以分配给数据链路的环数。如果底层物理链路上的 Rx 环数 (<i>rxringsavail</i>) 大于零, 则可以设置此值。 ■ <i>sw</i> 指示数据链路是基于软件的客户机。
<i>txringsavail</i>	只读	指示可以分配给物理数据链路上基于硬件的客户机的 Tx 环数。
<i>txhwclntavail</i>	只读	指示可以在物理数据链路上创建的基于硬件的 Tx 客户机数。
<i>txrings</i>	读写	指示数据链路独占使用的 Tx 环数。可以将此属性设置为三个可能的值之一： <ul style="list-style-type: none"> ■ <i>hw</i> 指示您正在配置基于硬件的客户机。如果底层物理链路上的基于硬件的 Tx 客户机数 (<i>txhwclntavail</i>) 大于零, 则可以设置此值。 ■ <i>number</i> 指示可以分配给数据链路的环数。如果底层物理链路上的 Tx 环数 (<i>txringsavail</i>) 大于零, 则可以设置此值。 ■ <i>sw</i> 指示数据链路是基于软件的客户机。

- # `dladm show-phys -H link`

显示现有客户机当前正在如何使用物理数据链路的环。

- # `dladm create-vnic -p ring-properties -l link VNIC`

`-p ring-properties` 指可以设置其值的环属性。

为客户机创建一定数量的 Rx 环或 Tx 环。

- # `dladm set-linkprop -p ring-properties VNIC`

如果环可用且支持环分配, 则将环分配给特定客户机。

显示数据链路上的环使用和环分配

要显示数据链路的 Rx 环和 Tx 环的可能值、已配置值和生效值, 请使用下面的命令语法：

```
# dladm show-linkprop -p rxrings,txrings link
```

要显示客户机当前正在如何使用物理数据链路的环, 请使用下面的命令语法：

```
# dladm show-phys -H link
```

例 7-1 数据链路上的环使用和环分配

以下示例说明了数据链路 net4 上的环分配。

```
# dladm show-linkprop net4
LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
...
net4      rxrings       rw  1              --             --           sw,hw,<1-7>
net4      txrings       rw  1              --             --           sw,hw,<1-11>
net4      txringsavail  r-  10             10            --           --
net4      rxringsavail  r-  7              7             --           --
net4      rxhwclntavail r-  3              3             --           --
net4      txhwclntavail r-  3              3             --           --
...
```

该输出显示数据链路 net4 独占使用一个 Rx 环和一个 Tx 环。数据链路 net4 具有七个 Rx 环和十个 Tx 环可供分配给客户机。可以在数据链路 net4 上创建三个基于硬件的 Rx 客户机和三个基于硬件的 Tx 客户机。

以下示例说明了数据链路 net0 的环使用。

```
# dladm show-phys -H net0
LINK  RINGTYPE  RINGS  CLIENTS
net0  RX        0-1    <default,mcast>
net0  TX        0-7    <default>net0
net0  RX        2-3    net0
net0  RX        4-5    --
net0  RX        6-7    --
```

从该输出来看，分配给 net0 的两个 Rx 环是环 2 和 3。对于 Tx 环，net0 使用环 0 到 7。

配置客户机并分配环

本节介绍了如何基于支持的环分配类型在数据链路上配置客户机。

▼ 如何配置客户机并分配环

请确保您可以解释显示数据链路环属性的 dladm 命令的输出，如[“用于配置环的命令” \[145\]](#)中所述。这些信息可帮助您配置客户机并分配环。

1. 成为管理员。
有关更多信息，请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“[使用所指定的管理权限](#)”。
2. 显示底层物理数据链路的属性。

```
# dladm show-linkprop -p rxringsavail,txringsavail,rxhwclntavail,txhwclntavail link
```

根据命令输出确定以下信息：

- NIC 是否支持基于硬件的客户机
- 要分配给基于硬件的客户机的环的可用性
- 您可以在链路上配置的基于硬件的客户机的可用性

3. 根据前一个步骤所获得的信息，执行以下操作之一：

- 使用下面的语法创建基于硬件的客户机：

```
# dladm create-vnic -p rxrings=value[,txrings=value] -l link VNIC
```

其中 *value* 可以为以下值之一：

- *hw* – 指示您正在配置基于硬件的客户机。
- *number* – 指示您只在配置基于硬件的客户机。*number* 指可分配给客户机以供其独占使用的环数。

- 使用下面的语法创建基于软件的客户机：

```
# dladm create-vnic -p rxrings=sw[,txrings=sw] -l link VNIC
```

另外，如果以前创建了客户机，则可以使用 `dladm set-linkprop` 命令设置环属性。

4. (可选) 验证您创建的客户机的环信息。

```
# dladm show-linkprop -p rxrings,txrings VNIC
```

5. (可选) 验证分布在不同客户机之间的链路环。

```
# dladm show-phys -H link
```

例 7-2 在 `nxge` 设备上配置客户机并分配环

此示例基于 `nxge` 设备，并说明了如何在数据链路 `net5` 上配置客户机并分配环。此示例说明了如何创建以下客户机：

- VNIC `vnic2`，是独占使用 Rx 环和 Tx 环的基于硬件的客户机。
- VNIC `vnic3`，是具有固定数量的环（根据 NIC 驱动程序的初始配置设置）的基于硬件的客户机。
- VNIC `vnic4`，是基于软件的客户机。

1. 检查物理数据链路 `net5` 是否支持客户机的环分配。

```
# dladm show-linkprop -p rxringsavail,txringsavail net5
LINK      PROPERTY      PERM VALUE      EFFECTIVE  DEFAULT  POSSIBLE
```

```
net5    rxringsavail  r-   7     7     --     --
net5    txringsavail  r-  11    11     --     --
```

该输出显示物理数据链路 net5 具有 7 个 Rx 环和 11 个 Tx 环，您可以将这些环指定给物理数据链路 net5 上的客户机。

2. 检查可在物理数据链路 net5 上创建的基于硬件的客户机的可用性。

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail net5
LINK  PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net5   rxhwclntavail  r-    3     3         --      --
net5   txhwclntavail  r-    4     4         --      --
```

该输出显示可以在数据链路 net5 上创建 3 个基于硬件的 Rx 客户机和 4 个基于硬件的 Tx 客户机。

3. 检查物理数据链路 net5 上的现有环使用情况。

```
# dladm show-phys -H net5
LINK  RINGTYPE  RINGS  CLIENTS
nxge1  RX        0-7    <default,mcast>
nxge1  TX        0-11   <default>
```

该输出显示 nxge1 设备具有八个 Rx 环 (0-7) 和十二个 Tx 环 (0-11)。由于在 nxge1 设备上无任何数据链路，因此没有将 Rx 环和 Tx 环指定给任何数据链路。CLIENTS 列中的 <default> 值表示 Tx 环将由基于软件的客户机使用。CLIENTS 列下的 <default,mcast> 值表示 Rx 环将由基于软件的客户机和非单点传送包使用。

4. 在数据链路 net5 上创建具有两个 Rx 环和两个 Tx 环的 VNIC vnic2。

```
# dladm create-vnic -l net5 -p rxrings=2,txrings=2 vnic2
```

5. 验证指定给 VNIC vnic2 的环。

```
# dladm show-linkprop -p rxrings,txrings vnic2
LINK  PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
vnic2  rxrings   rw    2     2         --      sw,hw,<1-7>
vnic2  txrings   rw    2     2         --      sw,hw,<1-11>
```

6. 验证物理数据链路 net5 上的环使用情况。

```
# dladm show-phys -H net5
LINK  RINGTYPE  RINGS  CLIENTS
nxge1  RX        0,3-7  <default,mcast>
nxge1  TX        0,3-11 <default>
nxge1  RX        1-2    vnic2
nxge1  TX        1-2    vnic2
```

该输出显示分配给 vnic2 的 Rx 环是 1 和 2。对于 Tx 环，vnic2 使用环 1 和 2。

7. 检查是否可以在物理数据链路 net5 上创建其他基于硬件的客户机。

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail net5
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net5	rxhwclntavail	r-	2	2	--	--
net5	txhwclntavail	r-	3	3	--	--

该输出显示可以在物理数据链路 net5 上创建两个基于硬件的 Rx 客户机和三个基于硬件的 Tx 客户机。

8. 创建 VNIC vnic3，它是基于硬件的客户机。

```
# dladm create-vnic -l net5 -p rxrings=hw,txrings=hw vnic3
```

9. 验证指定给 VNIC vnic3 的环。

```
# dladm show-linkprop -p rxrings,txrings vnic3
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
vnic3	rxrings	rw	--	1	--	sw, hw, <1-7>
vnic3	txrings	rw	hw	hw	--	sw, hw, <-11>

注 - 指定给客户机的环数取决于网络设备。将一个环指定到使您能够显式指定环数设备中的客户机，例如，nxge 设备。对于其他设备，指定给客户机的环数取决于设备的配置方式。请参见例 7-3 “在 ixgbe 设备上配置客户机并分配环”。

10. 检查是否可以在物理数据链路 net5 上创建其他基于硬件的客户机。

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail net5
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net5	rxhwclntavail	r-	2	2	--	--
net5	txhwclntavail	r-	2	2	--	--

该输出显示可以在物理数据链路 net5 上创建 2 个基于硬件的 Rx 客户机和 2 个基于硬件的 Tx 客户机。

11. 创建 VNIC vnic4，它是基于软件的客户机。

```
# dladm create-vnic -l net5 -p rxrings=sw,txrings=sw vnic4
```

12. 验证 vnic4 上的环使用情况。

```
# dladm show-linkprop -p rxrings,txrings vnic4
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
vnic4	rxrings	rw	sw	--	--	sw, hw, <1-7>
vnic4	txrings	rw	sw	--	--	sw, hw, <1-11>

13. 验证物理数据链路 net5 上的环使用情况。

```
# dladm show-phys -H net5
```

LINK	RINGTYPE	RINGS	CLIENTS
nxge1	RX	0,4-7	<default,mcast>,vnic4
nxge1	TX	0,4-11	<default>,vnic4
nxge1	RX	1-2	vnic2
nxge1	RX	3	vnic3

```

nxge1 TX 1-2 vnic2
nxge1 TX 3 vnic3

```

该输出显示 vnic4 是物理数据链路 net5 上共享缺省环集的基于软件的客户机。VNIC vnic2 是独占使用两个环 (2-3) 的基于硬件的客户机，而 vnic3 是独占使用一个环 (3) 的基于硬件的客户机。

例 7-3 在 ixgbe 设备上配置客户机并分配环

此示例基于 ixgbe 设备，并说明了如何在物理数据链路 net4 上配置客户机并分配环。

1. 检查物理数据链路 net4 上的现有环使用情况。

```

# dladm show-phys -H net4
LINK    RINGTYPE  RINGS    CLIENTS
net4    RX        0-3      <default,mcast>
net4    RX        4-7      --
net4    RX        8-11     --
net4    RX        12-15    --
net4    TX        0-7      <default>

```

2. 检查是否可以在物理数据链路 net4 上创建基于硬件的客户机。

```

# dladm show-linkprop -p rxhwcntavail,txhwcntavail,rxringsavail,txringsavail net4
LINK    PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net4    rxhwcntavail  r-    3      3          --      --
net4    txhwcntavail  r-    0      0          --      --
net4    rxringsavail  r-    0      0          --      --
net4    txringsavail  r-    0      0          --      --

```

该输出显示可以在物理数据链路 net4 上创建 3 个基于硬件的 Rx 客户机。

3. 创建 VNIC vnic3，它是基于硬件的 Rx 客户机。

```
# dladm create-vnic -l net4 -p rxrings=hw vnic3
```

无法为 vnic3 配置 txrings 属性，因为基于硬件的 Tx 客户机的可用数目 (txhwcntavail) 为零。

4. 验证指定给 VNIC vnic3 的环。

```

# dladm show-linkprop -p rxrings,txrings vnic3
LINK    PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
vnic3   rxrings    rw    hw     hw         --      sw,hw
vnic3   txrings    rw    --     8         --      --

```

5. 检查是否可以在物理数据链路 net4 上创建其他基于硬件的客户机。

```

# dladm show-linkprop -p rxhwcntavail,txhwcntavail,rxringsavail,txringsavail net5
LINK    PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net4    rxhwcntavail  r-    2      2          --      --

```

```
net4    txhwclntavail  r-    0      0      --      --
net4    rxringsavail  r-    0      0      --      --
net4    txringsavail  r-    0      0      --      --
```

该输出显示可以在物理数据链路 net4 上创建 2 个基于硬件的 Rx 客户机。

6. 验证物理数据链路 net4 上的环使用情况。

```
# dladm show-phys -H net4
LINK      RINGTYPE  RINGS    CLIENTS
net4      RX        0-3      <default,mcast>
net4      RX        4-7      vnic3
net4      RX        8-11
net4      RX        12-15    --
net4      TX        0-7      <default>,vnic3
```

该输出显示 vnic3 是独占使用四个环的基于硬件的 Rx 客户机。对于 Tx 环，vnic3 使用一组缺省环，并且与在物理数据链路 net4 上创建的其他数据链路共享这些环。

管理池和 CPU

在 Oracle Solaris 中，区域管理包括使用 `zonecfg` 或 `poolcfg` 命令为非网络进程指定 CPU 资源池。要指定同一个资源池还用来管理网络进程，请使用 `dladm set-linkprop` 命令来配置链路的 `pool` 属性。使用链路属性 `pool` 可以指定用于网络进程的 CPU 池。使用此属性，您可以更好地整合区域中的网络资源管理与 CPU 分配和管理。

通过设置链路的 `pool` 属性并将该链路指定为一个区域的网络接口，该链路将绑定到该区域的池。如果该区域设置为专用区域，则未指定给该区域的其他链路不能再使用池中的 CPU 资源。

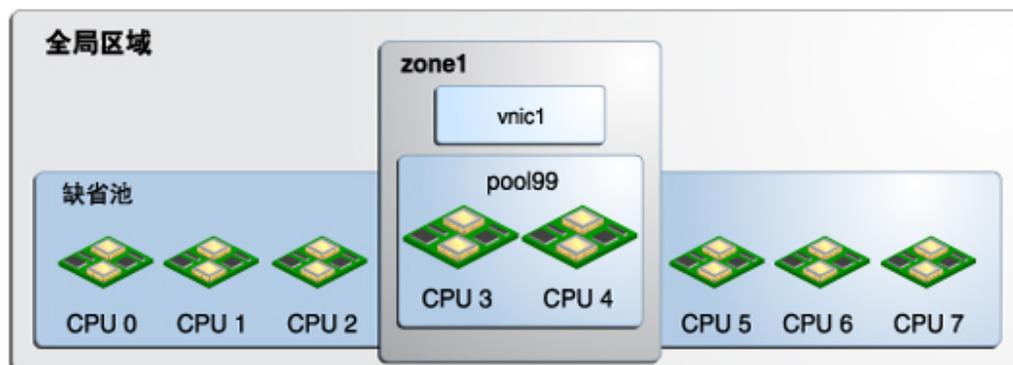
注 - 可以设置一个单独的属性 `cpus` 来将特定 CPU 指定给数据链路。`cpus` 和 `pool` 属性相互排斥。您不能为给定数据链路同时设置这两个属性。要使用 `cpus` 属性将 CPU 资源指定给数据链路，请参见[如何向数据链路分配 CPU \[156\]](#)。

有关区域内的池的更多信息，请参见《[在 Oracle Solaris 11.2 中进行资源管理](#)》中的第 13 章“[创建和管理资源池的任务](#)”。有关创建池以及将 CPU 集指定给池的更多信息，请参见 [poolcfg\(1M\)](#) 手册页。

使用池和 CPU

下图显示了将 `pool` 属性指定给数据链路时池的工作原理。

图 7-1 指定给一个区域的 VNIC 的 pool 属性

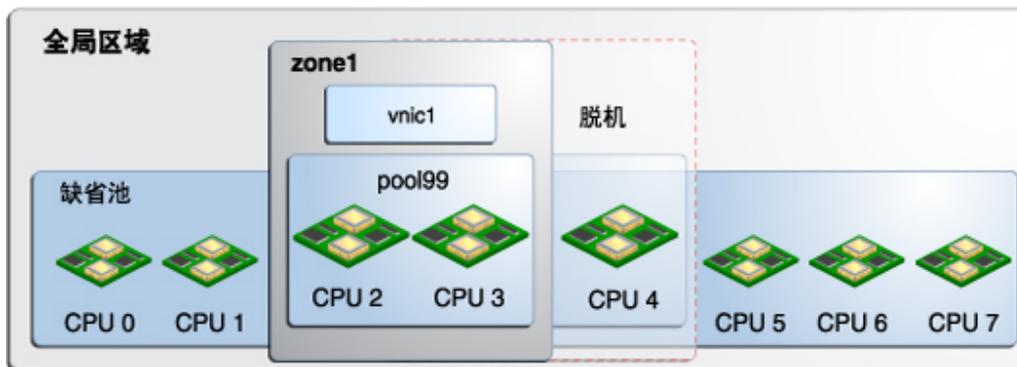


在图中，系统有八个 CPU。如果系统上没有配置池，则所有 CPU 都属于缺省池并由全局区域使用。不过，在本示例中，已创建 pool99 池，其中包含 CPU 3 和 CPU 4。该池与专用区域 zone1 相关联。如果 pool99 设置为 vnic1 的一个属性，则 pool99 变为专用的，还用来管理 vnic1 的网络进程。在将 vnic1 指定为 zone1 的网络接口后，pool99 中的 CPU 将被保留，用于管理 zone1 的网络和非网络进程。

pool 属性在本质上是动态的。可以使用一个 CPU 范围配置区域池，内核确定将哪些 CPU 指定给该池的 CPU 集。对池的更改会自动针对数据链路实现，从而简化了该链路的池管理。相反，通过使用 cpu 属性将特定 CPU 指定给链路时会要求您指定要分配的 CPU。每次要更改池的 CPU 组成时，您必须设置 cpu 属性。

例如，假定使图 7-1 “指定给一个区域的 VNIC 的 pool 属性” 中的系统 CPU 4 脱机。由于 pool 属性是动态的，软件会自动将另一个 CPU 与池相关联。因此，会保留池的两个 CPU 的初始配置。对于 vnic1，此更改是透明的。更新后的配置如下图所示。

图 7-2 pool 属性的自动重新配置



使用 `dladm show-linkprop` 命令显示数据链路的信息时，针对 `pool` 和 `cpus` 数据链路属性的 `EFFECTIVE` 列中的值指示这些属性当前的系统选定值。

针对 `pool` 和 `cpus` 属性显示了以下只读值：

- 对于 `pool` 数据链路属性，`EFFECTIVE` 列中的值指示用于网络处理的池。
- 对于 `cpus` 数据链路属性，`EFFECTIVE` 列中的值指示用于网络处理的 CPU。有关说明如何显示数据链路的 `cpus` 属性的示例，请参见例 7-5 “向数据链路分配 CPU”。

要管理区域的 CPU 资源，您不需要设置数据链路的池属性。您可以使用诸如 `zonecfg` 和 `poolcfg` 等命令将区域配置为使用某个资源池。如果没有为数据链路设置 `cpus` 和 `pool` 链路属性，数据链路的 `pool` 和 `cpus` 属性的 `EFFECTIVE` 列中的值是根据引导区域时区域的配置自动设置的。缺省池显示在 `pool` 属性的 `EFFECTIVE` 列中，系统将选择 `cpus` 属性的 `EFFECTIVE` 列中的值。因此，如果使用 `dladm show-linkprop` 命令，则 `pool` 和 `cpus` 属性的值为空，但值会显示在 `pool` 和 `cpus` 属性的 `EFFECTIVE` 列中。

您还可以直接设置数据链路的 `pool` 和 `cpu` 属性，以为联网进程分配区域的 CPU 池。配置这些属性后，它们的值会反映在 `pool` 和 `cpus` 属性的 `EFFECTIVE` 列中。但是，通常不使用此替代方法来管理区域的网络资源。

为数据链路配置 CPU 池

本节介绍了如何在创建数据链路时或者以后在数据链路需要进一步配置时，为链路设置 `pool` 属性。

▼ 如何为数据链路配置 CPU 池

开始之前 您必须已完成以下任务：

- 使用为其指定的 CPU 数量创建一个处理器集
- 创建一个将与处理器集相关联的池
- 将池与处理器集相关联

注 - 有关完成这些先决条件的说明，请参见《在 Oracle Solaris 11.2 中进行资源管理》中的“如何修改配置”。

1. 将链路的 `pool` 属性设置为您为区域创建的 CPU 池。

- 如果尚未创建 VNIC，请使用以下语法：

```
# dladm create-vnic -l link -p pool=pool vnic
```

- 如果 VNIC 存在，请使用以下语法：

```
# dladm set-linkprop -p pool=pool vnic
```

2. 设置区域以使用 VNIC。

```
global# zonecfg -z zone
zonecfg:zone> add net
zonecfg:zone:net> set physical=VNIC
zonecfg:zone:net> end
```

3. 验证并提交您已经实施的更改，然后退出该区域。

```
zonecfg:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

例 7-4 将链路的 CPU 池指定给区域

本示例说明如何将池指定给区域的数据链路。该方案基于图 7-1 “指定给一个区域的 VNIC 的 `pool` 属性”中的配置。本示例假定已为区域配置一个名为 `pool99` 的 CPU 池。然后将该池指定给一个 VNIC。最后，将非全局区域 `zone1` 设置为使用该 VNIC 作为网络接口。

```
# dladm create-vnic -l net1 -p pool=pool99 vnic1

# zonecfg -z zone1
```

```
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
```

向数据链路分配 CPU

本节介绍了如何通过配置 `cpu` 属性为数据链路指定 CPU 资源。与环不同的是，无法为数据链路分配专用 CPU。可以将同一组 CPU 分配给多个数据链路。

▼ 如何向数据链路分配 CPU

1. 成为管理员。

有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 验证为接口指定的 CPU。

```
# dladm show-linkprop -p cpus link
```

3. 将 CPU 指定给链路。

为数据链路处理包的 CPU 的列表。数据链路中断也可针对列表中的某个 CPU。

```
# dladm set-linkprop -p cpus=cpu1,cpu2,... link
```

`cpu1,cpu2,...` 指要指定给链路的 CPU 数目。您可以为链路指定多个专用 CPU。

4. (可选) 显示与链路相关联的 CPU。

```
# dladm show-linkprop -p cpus link
```

例 7-5 向数据链路分配 CPU

此示例说明了如何将特定 CPU 指定给数据链路 `net0`。

```
# dladm show-linkprop -p cpus net0
LINK    PROPERTY  PERM  VALUE      EFFECTIVE  DEFAULT  POSSIBLE
net0    cpus      rw    --         0-2        --       --
```

该输出显示系统已隐式将三个 CPU (0-2) 指定给数据链路 `net0`。但是，这些 CPU 并非唯一地分配给数据链路 `net0`。

```
# dladm set-linkprop -p cpus=0,1 net0
```

```
# dladm show-linkprop -p cpus net0
LINK      PROPERTY  PERM  VALUE      EFFECTIVE  DEFAULT  POSSIBLE
net0      cpus      rw    0-1        0-1        --       --
```

该输出显示您已显式将两个 CPU (0-1) 指定给数据链路 net0。分配的 CPU 将为数据链路 net0 处理包。

使用流管理网络资源

流是基于单个属性或属性组合对网络包进行分类的定制方式。流使您能够进一步指定网络资源。有关流的概述，请参见[“通过使用流管理网络资源” \[19\]](#)。

使用流管理网络资源涉及以下步骤：

1. 创建流。

流是基于派生自包头信息的单个属性或属性组合创建的。

可以使用下列属性之一将包通信组织到流中：

- 本地 IP 地址。
- 远程 IP 地址。
- 传输协议名称 (UDP、TCP 或 SCTP)。
- 区分服务字段 (Differentiated Services Field, DS Field) 属性，仅用于 IPv6 包中的 QoS。有关更多信息，请参见《[在 Oracle Solaris 11.2 中管理 IP 服务质量](#)》。

可以使用下列属性组合之一将包通信组织到流中：

- 传输协议名称 (UDP、TCP 或 SCTP) 和本地应用程序端口号 (例如，用于 FTP 的端口 21)。
- 传输协议名称 (UDP、TCP 或 SCTP) 和远程应用程序端口号。
- 传输协议名称 (UDP、TCP 或 SCTP) 和本地 IP 地址及本地应用程序端口号。此属性组合可以进一步包括远程 IP 地址和远程应用程序端口号：传输协议名称 (UDP、TCP 或 SCTP) + 本地 IP 地址 + 本地应用程序端口号 [+ 远程 IP 地址 [+ 远程应用程序端口号]]。

一个流只能基于这些属性组合之一。例如，您可以根据正在使用的传输协议和端口 (例如用于 FTP 的 TCP 端口 21) 或者根据 IP 地址 (例如来自特定源 IP 地址的包) 创建一个流。在最常见的情况下，您可以通过指定传输协议、本地或远程 IP 地址和本地或远程端口来创建一个流。另外，属于同一链路的所有流必须具有相同的属性组合。

2. 通过设置与网络资源相关的属性，定制流的资源使用。当前，带宽和优先级属性可以与流相关联。

有关更多信息，请参见[“配置流” \[158\]](#)。

用于流中的资源分配的命令

用于在流中分配网络资源的命令如下所述：

- 要创建一个流并同时向其添加资源，请使用以下命令语法：

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] -p prop=value[,...] flow
```

表明流特征的已定义属性集构成系统的流控制策略。有关可用于将包通信组织到流中的各种属性的列表，请参见“使用流管理网络资源” [157]。

- 要设置现有流的属性，请使用以下命令语法：

```
# flowadm set-flowprop -p prop=value[,...] flow
```

其中，*prop* 是指可指定给流的流属性。流属性与直接指定给链路的属性相同。但是，只有带宽和优先级属性可以与流相关联。要配置这些属性，请参见[如何配置流 \[158\]](#)。

有关更多信息，请参见 [flowadm\(1M\)](#) 手册页。

配置流

本节介绍了如何创建流并设置流属性。

▼ 如何配置流

1. 成为管理员。
有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. (可选) 列出可用链路以确定您将在其上配置流的链路。

```
# dladm show-link
```

3. 验证对所选链路上的 IP 接口正确配置了 IP 地址。

```
# ipadm show-addr
```

4. 根据您为每个流确定的属性创建流。

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] flow
```

link 指您在其上配置流的链路。

attribute 指可用于将网络包组织到流中的单个属性或属性组合。有关属性的信息，请参见“使用流管理网络资源” [157]。

flow 指您为流指定的名称。

有关流和流属性的更多信息，请参见 [flowadm\(1M\)](#) 手册页。

5. (可选) 显示链路的带宽值的可能范围。

```
# dladm show-linkprop -p maxbw link
```

link 指在其上配置流的数据链路。

值范围在命令输出的 POSSIBLE 字段下列出。

6. 通过设置适当的流属性在流上实现资源控制。

```
# flowadm set-flowprop -p prop=value[,...] flow
```

您可以指定以下用于控制资源的流属性：

maxbw 用流标识的包可以使用的最大链路带宽量。您设置的值必须在链路带宽值的允许范围内。

priority 处理属于指定流的包时将采用的优先级。priority 属性的允许值是 high、medium、low。如果某个流的优先级设置为 high，则属于该流的所有包将在同一链路上的其他包之前进行处理。此属性用于为对延迟敏感的应用程序创建流。此属性的缺省值为 medium。

注 - 目前，将 priority 属性从 medium 设置为 low 没有作用。

7. (可选) 显示已在数据链路上创建的流。

```
# flowadm
```

注 - 如果使用 flowadm 命令时不带任何子命令，该命令将提供与 flowadm show-flow 命令相同的信息。

8. (可选) 显示指定流的属性值。

```
# flowadm show-flowprop flow
```

此命令可显示 maxbw 和 priority 流属性以及只读的 hwflow 属性。

hwflow 一个只读属性，帮助您了解流中的包分类。此属性的可能值包括 on 和 off。值 on 表示流负载已移到 NIC，并且流的包分类是在硬

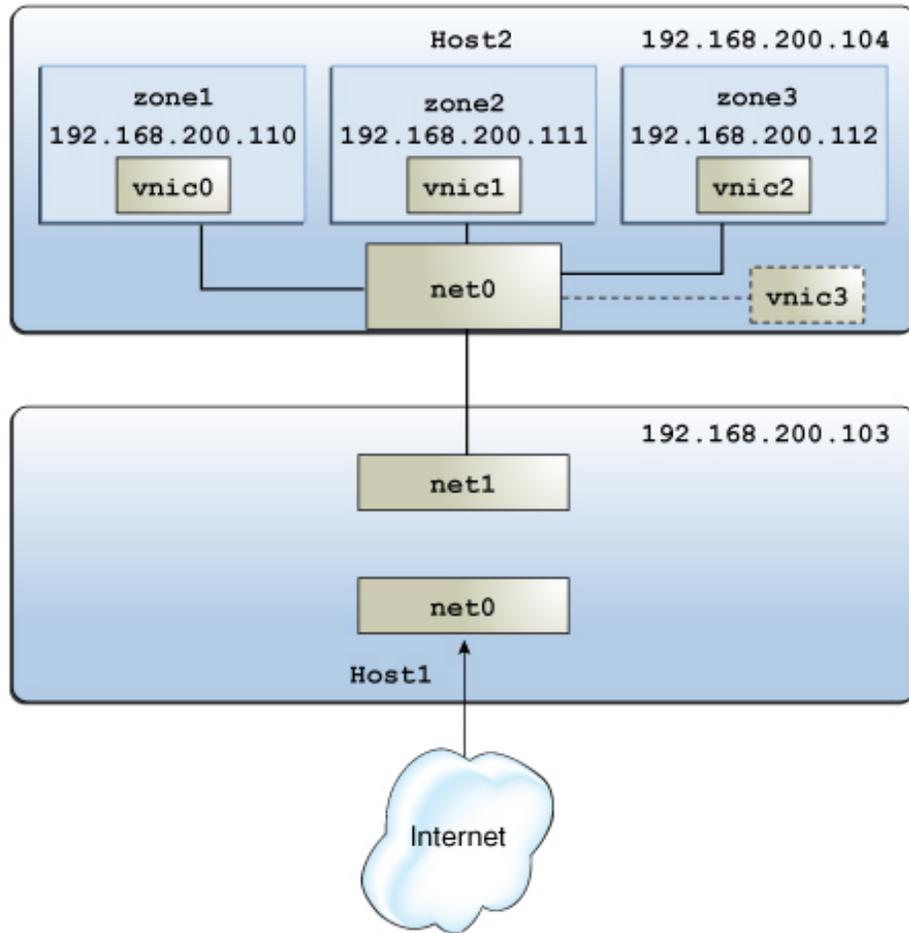
件级别进行的。此属性不能与 `flowadm add-flow`、`flowadm set-flowprop` 或 `flowadm reset-flowprop` 命令中的 `-p` 选项一起使用。

注 - 目前，只有对于通过指定所有的传输协议、本地或远程 IP 地址和本地或远程端口定义的流，才能为 `hwflow` 指定 `on` 值。另外，并非所有 NIC 都支持 `hwflow` 属性。

使用案例：通过设置数据链路和流属性来管理网络资源

下面的使用案例基于您通过设置数据链路和流属性来提高系统效率的方案。此使用案例基于下图中显示的配置。

图 7-3 用于管理数据链路和流上的资源的系统配置



此图显示了以下两个相互连接的物理主机：

- Host1 具有以下配置：
 - 一个充当服务器和路由器的非全局区域。为该区域指定了两个接口：net0 接口连接到 Internet，net1 接口连接到包括 Host2 在内的内部网络。
 - 在 net1 上配置了流以隔离通信并对属于这些流的包如何使用资源实施控制。有关配置流的信息，请参见[“使用流管理网络资源” \[157\]](#)。

- Host2 具有以下配置：
 - 三个非全局区域及其各自的 VNIC。这些 VNIC 是在 net0 上配置的，其 NIC 卡支持环分配。有关环分配的更多信息，请参见“[管理 NIC 环](#)” [144]。
 - 每个区域的网络处理负载是不同的。在此示例中，zone1 充当 HTTP 客户机。其余区域（zone2 和 zone3）充当尝试通过安全 shell (SSH) 协议访问 Host1 的 SSH 客户机。zone1 的网络通信高于 zone2 和 zone3 并且对时间不敏感。而 zone2 和 zone3 的网络通信级别低且对时间敏感。因此，要加快处理 zone2 和 zone3 的网络通信，您需要限制分配给 zone1 的网络通信的带宽。如果为 zone1 分配的带宽不受限制，则它将使用所有可用带宽。这将导致拒绝为 zone2 和 zone3 这两个其余区域分配带宽。
 - 将一个单独的 VNIC 配置为基于软件的客户机。有关 MAC 客户机的概述，请参见“[在 MAC 客户机中分配环](#)” [144]。

此使用案例中的任务涉及以下操作：

- 创建流和配置流控制 – 在 net1 上创建流，以便对属于由 Host1 的 net1 接收的流的包实施单独的资源控制。
- 为 Host2 上的 VNIC 配置网络资源属性 – 根据处理负载，为每个区域的 VNIC 配置一组专用环。还要配置一个单独的没有专用环的 VNIC 作为基于软件的客户机的示例。

注 - 此使用案例不包括任何区域配置过程。要配置区域，请参见《[创建和使用 Oracle Solaris 区域](#)》中的第 1 章“[如何规划和配置非全局区域](#)”。

1. 查看有关 Host1 上的链路和 IP 接口的信息。

```
# ipadm
NAME          CLASS/TYPE    STATE    UNDER    ADDR
lo0           loopback     ok       --        --
  lo0/v4      static       ok       --        127.0.0.1/8
  lo0/v6      static       ok       --        ::1/128
net1          ip           ok       --        --
  net1/v4    static       ok       --        192.168.200.103/24
net0          ip           ok       --        --
  net0/v4    static       ok       --        10.134.76.129/24
```

2. 在 Host1 的 net1 上创建以下流：

- httpflow – 包含 zone1 和 net1 之间的所有 HTTP 通信。


```
# flowadm add-flow -l net1 -a transport=tcp,local_ip=192.168.200.103,\
local_port=80,remote_ip=192.168.200.110 httpflow
```
- sshflow – 包含传入和传出 net1 的所有 SSH 通信。


```
# flowadm add-flow -l net1 -a transport=tcp,local_ip=192.168.200.103,\
local_port=22 sshflow
```

3. 在流上实现资源控制。

- 对于 httpflow，请将最大带宽设置为 500M。

```
# flowadm set-flowprop -p maxbw=500M httpflow
```

- 对于 sshflow，请将优先级设置为 high。

```
# flowadm set-flowprop -p priority=high sshflow
```

4. 验证有关所创建流的信息。

```
# flowadm
FLOW      LINK      PROTO  LADDR          LPORT  RADDR          RPORT  DSFLD
httpflow  net1      tcp    192.168.200.103  80     192.168.200.110  --     --
sshflow   net1      tcp    192.168.200.103  22     --              --     --
```

flowadm show-flowprop

```
FLOW      PROPERTY  PERM  VALUE  DEFAULT  POSSIBLE
httpflow  maxbw     rw    500    --        --
httpflow  priority  rw    medium medium    low,medium,high
httpflow  hwflow    r-    off    --        on,off
sshflow   maxbw     rw    --     --        --
sshflow   priority  rw    high   medium    low,medium,high
sshflow   hwflow    r-    off    --        on,off
```

有关输出的更多信息，请参见 [flowadm\(1M\)](#) 手册页。

5. 在 Host2 上，为每个区域在 net0 上配置 VNIC。

```
# dladm create-vnic -l net0 vnic0
```

```
# dladm create-vnic -l net0 vnic1
```

```
# dladm create-vnic -l net0 vnic2
```

6. 在每个 VNIC 上实现资源控制。

```
# dladm set-linkprop -p rxrings=4,txrings=4 vnic0
```

```
# dladm set-linkprop -p rxrings=2,txrings=2 vnic1
```

```
# dladm set-linkprop -p rxrings=1,txrings=1 vnic2
```

7. 将 VNIC 指定给其各自的区域。

```
# zonecfg -z zone1
```

```
# zonecfg:zone1> add net
```

```
# zonecfg:zone1:net> set physical=vnic0
```

```
# zonecfg:zone1:net> end
```

```
# zonecfg:zone1> commit
```

```
# zonecfg:zone1> exit
```

```
# zoneadm -z zone1 reboot
```

```
# zonecfg -z zone2
```

```
# zonecfg:zone2> add net
```

```
# zonecfg:zone2:net> set physical=vnic1
# zonecfg:zone2:net> end
# zonecfg:zone2> commit
# zonecfg:zone2> exit
# zoneadm -z zone2 reboot
```

```
# zonecfg -z zone3
# zonecfg:zone3> add net
# zonecfg:zone3:net> set physical=vnic2
# zonecfg:zone3:net> end
# zonecfg:zone3> commit
# zonecfg:zone3> exit
# zoneadm -z zone3 reboot
```

8. 创建一个与主接口 net0 共享环的基于软件的客户机。

```
# dladm create-vnic -p rxrings=sw,txrings=sw -l net0 vnic3
```

9. 假设将 pool1 (Host2 中的一组 CPU) 指定给了 zone1。指定同一个 CPU 池 pool1 还用来管理 zone1 的网络进程。

```
# dladm set-linkprop -p pool=pool1 vnic0
```

监视网络通信流量和资源使用情况

本章介绍了用于监视关于数据链路和流上的网络资源使用情况的网络统计数据的任务。可以在系统上配置网络记帐以将网络通信统计数据记录在日志文件中。这些统计数据可以帮助您分析资源分配以便您进行置备、整合和记帐。本章介绍了您可用于显示网络通信统计数据的两个命令：`dlstat` 和 `flowstat`。

本章包含以下主题：

- “[监视数据链路和流的网络通信统计数据的概述](#)” [165]
- “[用于监视网络通信统计数据的命令](#)” [167]
- “[显示链路的网络通信统计数据](#)” [167]
- “[显示流的网络通信统计数据](#)” [172]
- “[为网络通信配置网络记帐](#)” [175]

监视数据链路和流的网络通信统计数据的概述

包在流入或流出系统时会遍历一个路径。准确地说，数据包分别通过 NIC 的接收 (Rx) 环和传送 (Tx) 环接收和传送。来自这些环的传入包将被向上传递到网络栈以供进一步处理，而传出包则被发送到网络。

您可以组合和分配系统资源来管理网络通信。可以同时监视数据链路和流的接收端与传送端的网络流量统计数据。本章重点介绍数据链路和流的接收端网络通信统计数据。

通过设置数据链路属性，可以配置数据链路上的接收环、传送环和其他资源。根据数据链路上的网络流量，您可以为数据链路分配专用硬件环，以提高系统的包处理效率。例如，您可以为网络通信流量最大的数据链路分配更多环。有关如何为数据链路分配硬件环的更多信息，请参见“[配置客户机并分配环](#)” [147]。

由于以下原因，数据链路可能没有专用硬件环：

- 缺少硬件资源。例如，可能没有可专门指定给数据链路的可用环。
- 缺少硬件功能。例如，NIC 不会公开硬件环。
- 数据链路可能未与较低的硬件数据链路关联。例如，在 `etherstub` 上创建 VNIC 时。

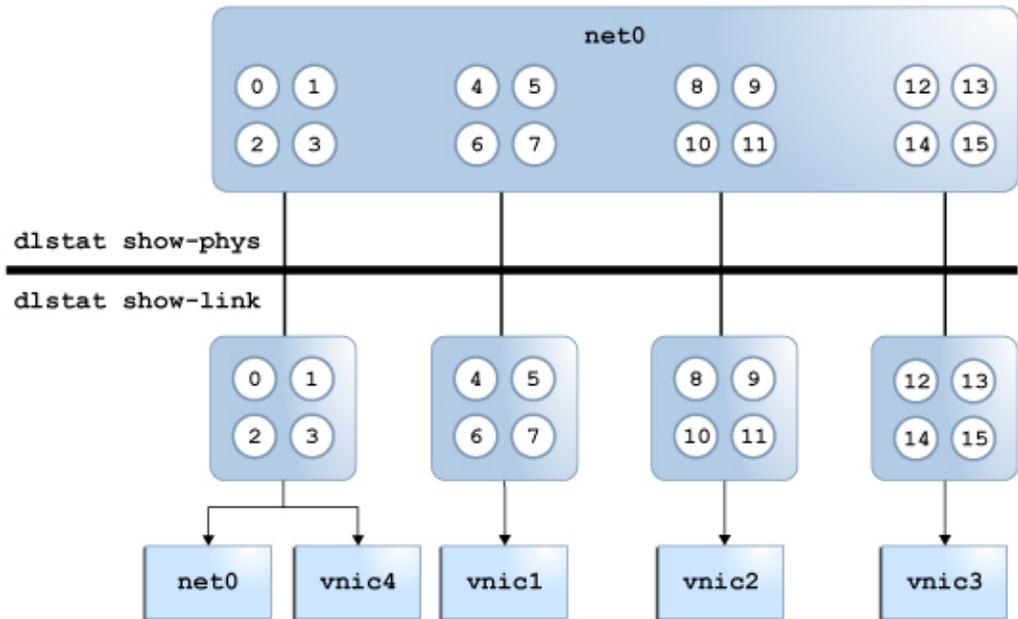
可能会因为以下原因而配置某些数据链路共享环：

- 数据链路可能不执行需要专用环的密集处理。

- NIC 可能不支持环分配。
- 尽管数据链路支持环分配，但环不再可供指定为专用环。

下图显示了数据链路中的硬件环分配情况。

图 8-1 数据链路中的环分配



下图显示了以下配置：

- 数据链路 net0 有 16 个可分配给其他数据链路的硬件环 (0-15)。
- 在数据链路 net0 上配置了 VNIC vnic1、vnic2、vnic3 和 vnic4。
- 为 VNIC vnic1、vnic2 和 vnic3 都分配了 4 个专用硬件环。
- 数据链路 net0 和 VNIC vnic4 共享硬件环 (0-3)。以下示例说明了物理数据链路 net0 的环分配。

```
# dladm show-phys -H net0
LINK      RINGTYPE  RINGS          CLIENTS
net0      RX         0-3            <default,mcast>,vnic4
net0      RX         4-7            vnic1
net0      RX         8-11           vnic2
net0      RX         12-15         vnic3
```

```
net0 TX 0-7 <default>,vnic4,vnic3,vnic2,vnic1
```

- 可以使用 `dlstat show-phys` 命令显示物理数据链路 `net0` 的网络通信统计数据。请参见例 8-1 “显示系统上的物理链路的通信统计数据”。
- 可以使用 `dlstat show-link` 命令显示数据链路 `net0`、`vnic1`、`vnic2`、`vnic3` 和 `vnic4` 的网络通信统计数据。请参见例 8-7 “显示具有专用硬件环的数据链路的网络通信统计数据”。

用于监视网络通信统计数据的命令

通过 `dlstat` 和 `flowstat` 命令，您可以分别监视数据链路和流的网络通信统计数据。以上命令等效于 `dladm` 和 `flowadm` 命令。下表将一对管理命令与一对监视命令的功能进行了比较。

管理命令		监视命令	
命令	功能	命令	功能
<code>dladm</code>	配置和管理数据链路	<code>dlstat</code>	显示数据链路的通信统计数据
<code>flowadm</code>	配置和管理流	<code>flowstat</code>	显示流的通信统计数据

显示链路的网络通信统计数据

您可以使用以下 `dlstat` 命令的变体显示网络通信信息。

命令	提供的信息
<code>dlstat [link]</code>	显示每个数据链路的传入和传出通信统计数据
<code>dlstat -rt [link]</code>	
<code>dlstat show-link [link]</code>	
<code>dlstat show-link -rt [link]</code>	显示每个数据链路每个环的传入和传出通信统计数据
<code>dlstat show-phys [link]</code>	显示每个网络物理设备的传入和传出通信统计数据
<code>dlstat show-phys -rt [link]</code>	显示每个网络物理设备每个环的传入和传出通信统计数据
<code>dlstat show-aggr [link]</code>	显示每个聚合每个端口的传入和传出通信统计数据
<code>dlstat show-aggr -rt [link]</code>	
<code>dlstat show-bridge [bridge]</code>	显示每个网桥的传入和传出通信统计数据
<code>dlstat show-bridge -rt [bridge]</code>	

您可以在 `dlstat` 命令中使用 `-r` 选项来显示接收端的统计数据，或使用 `-t` 选项来显示发送端的统计数据。有关其他选项的更多信息，请参见 [dlstat\(1M\)](#) 手册页。

显示网络设备的网络通信统计数据

`dlstat show-phys` 命令可提供有关物理网络设备的统计数据。如 [图 8-1 “数据链路中的环分配”](#) 所示，`dlstat show-phys` 命令作用于硬件环（位于网络栈的设备层）。

您可以使用下面的命令语法显示网络设备的网络通信统计数据：

```
# dlstat show-phys [-r|-t] [-Tu | -Td] [link] [interval [count]]
```

<code>-r</code>	只显示接收端的网络通信统计数据。不应该同时指定 <code>-t</code> 选项和此选项。 如果不指定 <code>-r</code> 选项或 <code>-t</code> 选项，则同时显示发送端和接收端的网络统计数据。
<code>-t</code>	只显示发送端的网络通信统计数据。不应该同时指定 <code>-r</code> 选项和此选项。 如果不指定 <code>-r</code> 选项或 <code>-t</code> 选项，则同时显示发送端和接收端的网络统计数据。
<code>-Tu</code>	以内部表示形式显示当前时间。
<code>-Td</code>	以标准日期格式显示当前时间。
<code>link</code>	要监视其网络统计数据的数据链路的名称。如果不指定数据链路，则将显示系统上配置的所有数据链路的相关信息。
<code>interval</code>	指定您希望刷新网络统计数据的时间（秒）。
<code>count</code>	指定您希望刷新所显示的网络通信统计数据的次数。如果不指定计数值，则将无限期刷新统计数据。

例 8-1 显示系统上的物理链路的通信统计数据

在此示例中，将同时显示系统上的每个链路的传入和传出网络通信。包的数量和每个包的字节大小已显示。

```
# dlstat show-phys
LINK   IPKTS   RBYTES   OPKTS   OBYTES
net5      0       0         0       0
net6      0       0         0       0
net0    25.57K   5.10M    1.93K   226.05K
net0      179     26.63K    161     22.75K
```

```

net3      0      0      0      0
net4      0      0      0      0
net2      0      0      0      0
net8      238  137.16K  191  8.41K
net1      0      0      0      0
...

```

该输出显示了以下信息：

LINK	物理或虚拟数据链路，由名称予以标识
IPKTS	链路上的传入包的数量
RBYTES	链路上接收的字节数
OPKTS	链路上的传出包的数量
OBYTES	此链路上发送的字节数

例 8-2 显示网络设备的接收端通信统计数据

在此示例中，以下列方式显示接收的网络通信统计数据：时间间隔值为 2 秒，计数值为 3。

```

# dlstat show-phys -r 2 3
LINK  TYPE  INDEX  IPKTS  RBYTES
net0  rx    0      8.03M  12.09G
net1  rx    0      0      0
net0  rx    0      8.79K  13.28M
net1  rx    0      0      0
net0  rx    0      8.50K  12.83M
net1  rx    0      0      0

```

将数据链路 net0 和 net1 视为一个组。第一组数据链路 net0 和 net1 显示接收的包和字节的总数。在此示例中，8.03M 是指接收的包总数，12.09G 是指 net0 接收的字节总数。第二组数据链路 net0 和 net1 以每秒速率的形式显示网络通信统计数据，也称为标准化值。即，8.79K 是指 net0 接收的包数的标准化值，采用的时间间隔为 2 秒。同样，第三组数据链路 net0 和 net1 也显示网络通信统计数据的标准化值，其时间间隔为 2 秒。

例 8-3 显示网络设备的接收端通信统计数据

在此示例中，显示了数据链路 net0 的传入通信统计数据。

```

# dlstat show-phys -r net0
LINK  TYPE  ID  INDEX  IPKTS  RBYTES
net0  rx    local  --    0      0
net0  rx    hw    1      0      0
net0  rx    hw    2      1.73M  2.61G
net0  rx    hw    3      0      0

```

```
net0    rx    hw    4    8.44M    12.71G
net0    rx    hw    5    5.68M    8.56G
net0    rx    hw    6    4.99M    7.38G
net0    rx    hw    7        0        0
```

在此示例中，net0 数据链路具有八个接收环，INDEX 字段下标识了这些环。在每个环上平均分布包是一种理想配置，表示根据链路的负载将环适当地分配到了各个链路。不均匀的分布表示每个链路的环分布不成比例。不均匀分布的解决方法取决于 NIC 是否支持动态环分配。如果 NIC 支持动态环分配，您可以重新分布每个链路上的环以均衡包处理负载。有关更多信息，请参见“[管理 NIC 环](#)” [144]。

例 8-4 显示网络设备的传送端通信统计数据

在此示例中，显示 net0 作为网络设备时其传送环的使用情况。

```
# dlstat show-phys -t net0
LINK  TYPE  INDEX  OPKTS  OBYTES
net0   tx    0      93     4.63K
net0   tx    1       0       0
net0   tx    2       0       0
net0   tx    3       0       0
net0   tx    4       0       0
net0   tx    5     47    11.02K
net0   tx    6     23     7.13K
net0   tx    7       0       0
```

例 8-5 在使用时间的情况下显示网络设备的通信统计数据

以下示例显示了 net0 作为网络设备时其网络通信的相关统计数据以及当前时间的内部表示形式。

```
# dlstat show-phys -Tu net0
1401652481
      LINK  IPKTS  RBYTES  OPKTS  OBYTES
net0   184   27.14K  165    22.91K
```

以下示例显示了 net0 作为网络设备时其网络通信的相关统计数据以及以标准日期格式表示的当前时间。

```
# dlstat show-phys -Td net0
Sun Jun 1 12:54:47 PDT 2014
      LINK  IPKTS  RBYTES  OPKTS  OBYTES
net0   184   27.14K  165    22.91K
```

显示数据链路的网络通信统计数据

可以使用 `dlstat show-link` 命令显示数据链路的网络通信统计数据。

例 8-6 显示数据链路的网络通信统计数据

此示例显示数据链路 vnic0 的网络通信统计数据。

```
# dlstat show-link vnic0
LINK  IPKTS  RBYTES  OPKTS  OBYTES
vnic0  3      180     0      0
```

例 8-7 显示具有专用硬件环的数据链路的网络通信统计数据

此示例显示具有 4 个专用 Rx 环的数据链路 vnic0 的接收端网络通信统计数据。该输出中 ID 列中的 hw 值指示数据链路 vnic0 有专用硬件环。

```
# dlstat show-link -r vnic0
LINK  TYPE   ID  INDEX  IPKTS  RBYTES  INTRS  POLLS  IDROPS
vnic0  rx    local  --      0      0      0      0      0
vnic0  rx    other  --      64    2.94K   0      0      0
vnic0  rx    hw     8       0      0      0      0      0
vnic0  rx    hw     9      53    7.97K   53     0      0
vnic0  rx    hw    10      4     392     4      0      0
vnic0  rx    hw    11  153.65K 220.68M 153.65K 0      0
```

例 8-8 显示数据链路的传送端网络通信统计数据

此示例显示数据链路 vnic0 的传送端网络通信统计数据。

```
# dlstat show-link -t vnic0
LINK  TYPE   ID  INDEX  OPKTS  OBYTES  ODROPS
vnic0  tx    local  --      0      0      0
vnic0  tx    other  --      19     798     0
vnic0  tx    sw     --      0      0      0
```

例 8-9 显示没有专用硬件环的数据链路的网络通信统计数据

此示例显示没有专用 Rx 环的数据链路 net6 的网络通信统计数据。该输出中 ID 列中的 sw 值指示数据链路 net6 没有配置专用硬件环。

```
# dlstat show-link -r net6
LINK  TYPE   ID  INDEX  IPKTS  RBYTES  INTRS  POLLS  IDROPS
net6  rx    local  --      0      0      0      0      0
net6  rx    other  --      0      0      0      0      0
net6  rx    sw     --      0      0      0      0      0
```

显示链路聚合的网络通信统计数据

dlstat show-aggr 命令显示了当通信遍历系统上的聚合时，每个聚合端口的网络包统计数据。

例 8-10 显示链路聚合的网络通信统计数据

```
# dlstat show-aggr
LINK      PORT  IPKTS  RBYTES  OPKTS  OBYTES
aggr0     --    13     832     13     780
aggr0     net0   0       0       13     780
aggr0     net3   13     832     0       0
```

在此示例中，输出指明了包含 net0 和 net3 这两个底层链路的链路聚合 aggr0 的配置。由于网络通信是系统通过聚合接收或发送的，因此将报告每个端口的有关传入和传出包以及它们各自大小的信息。端口通过聚合的底层链路标识。

有关链路聚合的信息，请参见《在 Oracle Solaris 11.2 中管理网络数据链路》中的第 2 章“使用链路聚合配置高可用性”。

显示网桥的网络通信统计数据

dlstat show-bridge 命令可显示每个网桥的网络统计数据并列出连接到每个网桥的链路的统计数据。

例 8-11 显示网桥的网络通信统计数据

在此示例中，显示了网桥 rbblue0 和 stbred0 的网络统计数据。

```
# dlstat show-bridge
BRIDGE    LINK  IPKTS  RBYTES  OPKTS  OBYTES  DROPS  FORWARDS
rbblue0   --    1.93K  587.29K  2.47K  3.30M   0       0
          simblue1  72    4.32K   2.12K  2.83M   0       --
          simblue2 1.86K  582.97K  348    474.04K 0       --
stbred0   --    975    976.69K  3.44K  1.13M   0       38
          simred3   347    472.54K  1.86K  583.03K 0       --
          simred4   628    504.15K  1.58K  551.51K 0       --
```

显示流的网络通信统计数据

流统计数据有助于评估系统上定义的所有流的包通信。要显示流统计数据，请使用 flowstat 命令。有关更多信息，请参见 [flowstat\(1M\)](#) 手册页。

可以使用下面的命令语法显示流的网络通信统计数据：

```
# flowstat [-r|-t] [-l link] [-Tu | -Td] [flow] [interval [count]]
```

-r 只显示接收端的网络通信统计数据。不应该同时指定 -t 选项和此选项。

	如果不指定 <code>-r</code> 选项或 <code>-t</code> 选项，则同时显示传送端和接收端的网络统计数据。
<code>-t</code>	只显示传送端的网络通信统计数据。不应该同时指定 <code>-r</code> 选项和此选项。 如果不指定 <code>-r</code> 选项或 <code>-t</code> 选项，则同时显示传送端和接收端的网络统计数据。
<code>-l link</code>	要监视其网络统计数据的数据链路的名称。如果不指定数据链路，则将显示系统上配置的所有流的相关信息。
<code>-Tu</code>	以内部表示形式显示当前时间。
<code>-Td</code>	以标准日期格式显示当前时间。
<code>flow</code>	要监视其网络统计数据的流的名称。如果不指定流，则根据指定的链路，显示所有流统计数据。
<code>interval</code>	指定您希望刷新网络统计数据的时间（秒）。如果不指定时间间隔值，则显示包和字节的总数。
<code>count</code>	指定您希望刷新所显示的网络通信统计数据的次数。如果不指定计数值，则将无限期刷新统计数据。

以下示例说明了显示系统中已配置流的信息的不同方法。

例 8-12 显示流的网络通信统计数据

在此示例中，显示了系统上配置的所有流的网络通信统计数据，时间间隔值为 1 秒，计数值为 2。

```
# flowstat 1 2
FLOW      IPKTS  RBYTES  IDROPS   OPKTS   OBYTES  ODROPS
flow1     1.78M  2.68G   443      889.57K 58.72M   0
flow2      0       0       0         0         0         0
flow1     8.31K  12.51M  243      4.22K   280.45K  0
flow2      0       0       0         0         0         0
```

将流 `flow1` 和 `flow2` 视为一个组。第一组流 `flow1` 和 `flow2` 显示流接收和传送的网络通信统计数据的总数。在此示例中，1.78M 是指 `flow1` 接收的包的总数。第二组流 `flow1` 和 `flow2` 以每秒速率的形式显示网络统计数据，也称为标准化值。在此示例中，8.31K 是指 `flow1` 接收的包数的标准化值，采用的时间间隔为 1 秒。

例 8-13 显示流的传送端通信统计数据

在此示例中，显示了有关在系统上配置的所有流的传出通信的网络通信统计数据。

```
# flowstat -t
FLOW      OPKTS      OBYTES      ODROPS
flow1     24.37M     1.61G       0
flow2      0          0           0
```

例 8-14 显示数据链路上的流的接收端通信统计数据

在此示例中，显示了数据链路 net0 上配置的所有流的传入网络通信，时间间隔值为 2 秒，计数值为 5。

```
# flowstat -r -l net0 2 5
FLOW      IPKTS      RBYTES      IDROPS
flow1     2.38M      3.59G       14.89K
flow2      0          0           0
flow1     8.24K      12.40M      180
flow2      0          0           0
flow1     8.94K      13.47M      206
flow2      0          0           0
flow1     7.43K      11.19M      161
flow2      0          0           0
flow1     8.38K      12.62M      213
flow2      0          0           0
```

将流 flow1 和 flow2 视为一个组。第一组流 flow1 和 flow2 显示流接收的包和字节的总数。在此示例中，2.38M 是指接收的包总数，3.59G 是指 flow1 接收的字节总数。第二组流 flow1 和 flow2 以每秒速率的形式显示网络统计数据，也称为标准化值。在此示例中，8.24K 是指 flow1 接收的包数的标准化值，采用的时间间隔为 2 秒。同样，后续的流组也显示网络通信统计数据的标准化值，定期时间间隔为 2 秒。

例 8-15 在使用时间的情况下显示流的通信统计数据

以下示例显示了在数据链路 net0 上创建的所有流的传入通信统计数据以及当前时间的内部表示形式。

```
# flowstat -r -l net0 -Tu
1364380279
      FLOW      IPKTS      RBYTES      IDROPS
      tcp-flow  183.11K   270.24M     0
      udp-flow   0         0           0
```

以下示例显示了在数据链路 net0 上创建的所有流的传入通信统计数据以及以标准日期格式表示的当前时间。

```
# flowstat -r -l net0 -Td
Wednesday, March 27, 2013 04:01:011 PM IST
      FLOW      IPKTS      RBYTES      IDROPS
      tcp-flow  183.11K   270.24M     0
      udp-flow   0         0           0
```

为网络通信配置网络记帐

您可以使用扩展记帐功能在系统上设置网络记帐。网络记帐涉及在日志文件中捕获有关网络通信的统计数据。您可以维护通信的记录以便进行跟踪、置备、整合和记帐。以后，您可以查看日志文件获取有关一段时间内网络使用情况的历史信息。

要设置网络记帐，请使用扩展记帐工具的 `acctadm` 命令。有关更多信息，请参见 [acctadm\(1M\)](#) 手册页。设置完网络记帐之后，请使用 `flowstat` 命令记录通信统计数据。

▼ 如何设置网络记账

1. 成为管理员。
有关更多信息，请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 查看扩展记帐功能可以支持的记帐类型的状态。

```
# acctadm [process | task | flow | net]
```

扩展记帐功能可以支持四种记帐类型。acctadm 命令的可选操作数对应于以下记帐类型：

- process – 进程记帐
- task – 任务记帐
- flow – 流记帐
- net – 网络记帐

注 - 网络记帐也适用于由“使用流管理网络资源” [157]中所述的 `flowadm` 和 `flowstat` 命令管理的流。因此，要设置这些流的记帐，请结合使用 `net` 选项和 `acctadm` 命令。请勿使用 `flow` 选项，该选项启用 IPQoS 配置的流记帐。

指定 `net` 会显示网络记帐的状态。如果不使用 `net`，则显示所有四种记帐类型的状态。

3. 为网络通信启用扩展记帐。

```
# acctadm -e extended -f filename net
```

其中 `filename` 包括用于捕获网络通信统计数据的日志文件的完整路径。可以在您指定的任何目录中创建日志文件。

4. 验证已激活的扩展网络记帐。

```
# acctadm net
```

例 8-16 在系统中设置网络记帐

此示例说明了如何在系统中配置网络记帐以捕获和显示通信历史信息。

查看所有记帐类型的状态，如下所示：

```
# acctadm
      Task accounting: inactive
      Task accounting file: none
      Tracked task resources: none
      Untracked task resources: extended
      Process accounting: inactive
      Process accounting file: none
      Tracked process resources: none
      Untracked process resources: extended,host
      Flow accounting: inactive
      Flow accounting file: none
      Tracked flow resources: none
      Untracked flow resources: extended
      Net accounting: inactive
      Network accounting file: none
      Tracked Network resources: none
      Untracked Network resources: extended
```

输出显示网络记帐未处于活动状态。因此，应启用扩展网络记帐。

```
# acctadm -e extended -f /var/log/net.log net
# acctadm net
      Net accounting: active
      Net accounting file: /var/log/net.log
      Tracked net resources: extended
      Untracked net resources: none
```

显示网络通信的历史统计数据

启用网络记帐之后，您可以使用 `dlstat` 和 `flowstat` 命令从日志文件中提取信息。

您必须为网络启用扩展记帐，才能显示有关网络的历史数据。此外，要显示有关流的通信历史数据，必须先按[“使用流管理网络资源” \[157\]](#)中所述在系统中配置流。

显示数据链路的网络通信的历史统计数据

您可以使用下面的命令语法显示数据链路的网络通信的历史统计数据：

```
# dlstat show-link -h [-a] -f filename [-d date] [-F format] [-s start-time] [-e end-time]
[link]
```

-h	按照数据链路的传入和传出包，显示有关资源使用情况的历史信息摘要。
-a	显示所有数据链路的资源使用情况，包括那些在数据捕获后已被删除的数据链路。
-f filename	指定在使用 acctadm 命令启用网络记帐时定义的日志文件。
-d date	显示指定日期的记录信息。
-F format	以特定格式显示数据，以便随后可以对其进行绘图分析。当前，gnuplot 是唯一受支持的格式。
-s start-time	指定要显示的所记录网络统计数据信息的开始时间。使用 MM/DD/YYYY, hh:mm:ss 格式。hour (hh) 必须使用 24 小时制表示法。如果未包含日期，则将显示对应于当前日期在指定时间范围内的数据。
-e end-time	指定要显示的所记录网络统计数据信息的结束时间。使用 MM/DD/YYYY, hh:mm:ss 格式。hour (hh) 必须使用 24 小时制表示法。如果未包含日期，则将显示对应于当前日期在指定时间范围内的数据。
link	显示指定数据链路的历史数据。如果不使用此选项，则显示所有配置的数据链路的历史网络数据。

例 8-17 显示有关数据链路的资源使用情况的历史统计数据

在此示例中，显示了有关系统中所有数据链路的网络通信及其资源使用情况的历史统计数据。

```
# dlstat show-link -h -f /var/log/net.log
LINK DURATION IPKTS RBYTES OPKTS OBYTES BANDWIDTH
net0 80 1031 546908 0 0 2.44 Mbps
net1 100 2045 235977 0 0 9.67 Mbps
```

显示流的网络通信历史统计数据

您可以使用下面的命令语法显示流的网络通信的历史统计数据：

```
# flowstat -h [-a] -f filename [-d date] [-F format] [-s start-time] [-e end-time] [flow]
```

-h	按照配置的流的传入和传出包，显示有关资源使用情况的历史信息摘要。
----	----------------------------------

- a 显示所有配置的流的资源使用情况，包括那些在数据捕获后已被删除的配置流。
- f *filename* 指定在使用 `acctadm` 命令启用网络记帐时定义的日志文件。
- d 显示指定日期的记录信息。
- F *format* 以特定格式显示数据。当前，`gnuplot` 是唯一受支持的格式。
- s *start-time* 指定要显示的所记录网络统计数据信息的开始时间。使用 `MM/DD/YYYY, hh:mm:ss` 格式。hour (hh) 必须使用 24 小时制表示法。如果未包含日期，则将显示对应于当前日期在指定时间范围内的数据。
- e *end-time* 指定要显示的所记录网络统计数据信息的结束时间。使用 `MM/DD/YYYY, hh:mm:ss` 格式。hour (hh) 必须使用 24 小时制表示法。如果未包含日期，则将显示对应于当前日期在指定时间范围内的数据。
- flow* 显示指定流的历史数据。如果不使用此选项，则显示所有配置的流的历史网络数据。

例 8-18 显示有关流的资源使用情况的历史统计数据

下面的示例显示了有关系统中流的通信资源使用情况的历史统计数据。

```
# flowstat -h -f /var/log/net.log
FLOW      DURATION  IPACKETS  RBYTES    OPACKETS  OBYTES    BANDWIDTH
flowtcp   100      1031      546908    0          0          43.76Kbps
flowudp   0         0         0          0          0          0.00Mbps
```

下面的示例显示了在给定的日期和时间范围内有关 `flowtcp` 的通信资源使用情况的历史统计数据。

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-f /var/log/net.log flowtcp

FLOW      START      END          RBYTES  OBYTES    BANDWIDTH
flowtcp   10:39:06  10:39:26   1546    6539     3.23 Kbps
flowtcp   10:39:26  10:39:46   3586    9922     5.40 Kbps
flowtcp   10:39:46  10:40:06   240     216      182.40 bps
flowtcp   10:40:06  10:40:26   0        0         0.00 bps
```

下面的示例通过使用 `gnuplot` 格式，显示了在给定的日期和时间范围内有关 `flowtcp` 的通信资源使用情况的历史统计数据。

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-F gnuplot -f /var/log/net.log flowtcp
# Time tcp-flow
10:39:06 3.23
10:39:26 5.40
```

10:39:46 0.18
10:40:06 0.00

索引

A

安装

- 边缘虚拟桥接, 71
- acctadm 命令, 175

B

- 本地 IP 地址, 157
- 本地应用程序端口号, 157
- 边缘控制协议 (edge control protocol, ECP), 77
 - 显示 ECP 状态和统计数据, 80
- 边缘虚拟桥接, 67 见 EVB
 - EVB 如何提高网络和服务器效率, 68
 - Oracle VSI 管理器, 77
 - oracle_v1, 77
 - oracle_v1 编码, 77
 - VDP 协议, 77
 - VNIC 信息交换的工作方式, 78
 - VSI 标识符, 77
 - VSI 版本, 77, 80
 - VSI 管理器, 77
 - VSI 管理器 ID, 77
 - VSI 类型 ID, 77, 80
 - VSI 配置文件, 77
- 交换 VNIC 信息, 77
- 以太网链路的 VDP 状态, 79
- 使用 EVB 时和未使用 EVB 时的服务器效率, 70
- 使用 LLDP 管理 VM 之间的通信, 76
- 使用 VDP, 77
- 允许 VM 通过外部交换机通信, 72
- 反射中继, 67, 68
- 安装, 71
- 控制 VM 中的通信, 71
- 数据链路属性, 80
- 显示 VDP 状态的示例, 79

- 显示 VDP 统计数据的示例, 79
- 显示链路属性的示例, 79
- 更改缺省 EVB 配置, 80
- 概述, 67
- 组件, 77
- 自动虚拟端口配置, 68
- 访问控制列表, 68

C

- 传送环 见 Tx 环
- 创建
 - IP 接口, 23
- 创建 VF VNIC, 46
- CPU, 18, 152
 - CPU 分配, 156
- CPU 池, 152
 - 使用, 152
 - 指定给链路, 155
 - 数据链路, 18
 - 缺省池, 153

D

- 带宽
 - 数据链路上的设置, 18
 - 流上的设置, 18
 - 流的设置, 159
- 单根 I/O 虚拟化 见 SR-IOV
- 第 2 层网络, 52
- DCBX 协议, 68
- dladm 命令, 94, 97
 - create-etherstub, 22, 32
 - create-vlan, 145
 - create-vnic, 21, 97, 144, 155

- create-vxlan , 57
- delete-vnic , 42
- delete-vxlan , 62
- modify-vnic , 38 , 39 , 41
- set-linkprop , 24 , 144
- show-ether , 79
- show-link , 61
- show-linkprop , 45 , 145 , 146
- show-phys , 48 , 146 , 146
- show-vnic , 34 , 36 , 48 , 97
- show-vxlan , 57 , 61

dlstat 命令 , 22 , 167 , 167

- show-aggr , 171
- show-bridge , 172
- show-link , 176
- show-phys , 168

dlstat show-ether 命令 , 79

DS 字段 , 157

E

etherstub , 13 , 14

- 专用虚拟网络 , 31
- 创建 , 22
- 配置 , 22

EVB , 17 见 边缘虚拟桥接

EVS

- 使用区域 , 99
- 名称空间管理 , 90
- 安全要求 , 99
- 必需软件包 , 98
- 益处 , 88
- 管理任务 , 101

EVS 管理器

- 描述 , 92

EVS 节点

- 定义 , 94

EVS 客户机 , 94

EVS 控制器 , 92

- 创建和管理 , 103
- 必需的软件包 , 99
- 必需软件包 , 103
- 显示 , 110
- 显示属性 , 110
- 规划 , 103

- 设置 , 110
- 设置属性 , 110
- 配置 , 105

EVS 控制器属性 , 92

- 显示 , 105
- 设置 , 105

EVS 组件 , 90

- EVS 客户机 , 90
- EVS 控制器 , 90
- EVS 管理器 , 90
- EVS 节点 , 90

evsadm

- 用于管理 EVS 客户机属性的子命令 , 96
- 用于管理 EVS 控制器属性的子命令 , 96
- 用于管理 IPnet 的子命令 , 95
- 用于管理 VPort 的子命令 , 95
- 用于管理弹性虚拟交换机的子命令 , 95

evsadm 命令 , 94 , 94

- add-ipnet , 95 , 114
- add-vport , 95 , 114
- create-evs , 95 , 114
- delete-evs , 95 , 127
- remove-ipnet , 95 , 121
- remove-vport , 95 , 126
- reset-vport , 96 , 126
- set-controlprop , 96 , 110
- set-evsprop , 95 , 95 , 119
- set-prop , 92 , 92 , 96 , 110
- set-vportprop , 123
- show-controlprop , 96 , 110
- show-evs , 95 , 118
- show-evsprop , 95 , 120
- show-ipnet , 95 , 122
- show-prop , 96 , 110
- show-vport , 95 , 125
- show-vportprop , 96 , 124

evsstat 命令 , 128

- 显示弹性虚拟交换机的网络通信统计数据 , 96

F

反射中继 , 14 , 67

访问控制列表 (access control list, ACL) , 68

服务级别协议 见 SLA

服务质量 (quality of service, QoS) , 18 , 143

flowadm 命令, 157, 159
 add-flow, 158, 158, 162
 help, 158
 set-flowprop, 158, 159, 163
 show-flowprop, 159, 163
 flowstat 命令, 22, 167, 172

G

更改
 缺省 EVB 配置, 80
 管理
 弹性虚拟交换机, evsadm 子命令, 95
 EVS 客户机属性, evsadm 子命令, 96
 EVS 控制器属性, evsadm 子命令, 96
 IPnet, evsadm 子命令, 95
 VPort, evsadm 子命令, 95
 管理 CPU, 152
 管理环
 命令, 145
 管理网络数据链路
 EVB, 17
 管理网络资源, 143
 使用数据链路属性, 143
 使用流, 157
 GARP VLAN 注册协议 (GARP VLAN Registration Protocol, GVRP), 24

H

环分配, 144, 144
 在 VLAN 上, 145
 环使用和环分配, 146

I

IP 地址, 89
 IP 网络 见 IPnet
 ipadm 命令, 162
 create-addr, 23
 create-ip, 23
 show-addr, 57
 显示 IP 信息, 162
 IPnet

删除, 121
 显示, 122
 添加到弹性虚拟交换机, 114
 管理, 117, 121

IPOIB

在区域中临时创建, 31

J

基于软件的客户机, 144, 144
 基于硬件的客户机, 144
 监视
 网络使用情况, 165
 网络通信统计数据
 命令, 167
 将在 VXLAN 上创建的 VNIC 分配给区域的示例, 61
 接收环 见 Rx 环

K

控制 VM 中的通信, 71
 扩展计帐工具
 链路和流的网络记帐, 175
 扩展记帐工具, 175
 任务记帐, 175
 流记帐, 175
 进程记帐, 175

L

临时创建 VNIC
 在区域中, 30
 流, 19, 22, 157
 优先级设置, 158
 创建, 158, 158
 基于属性, 157
 带宽设置, 158, 159
 显示信息, 159
 显示网络通信的历史统计数据, 177
 设置属性, 158
 配置, 158
 流控制, 157
 LLDP, 68
 使用 LLDP 管理 VM 之间的通信, 76

M

命令

- 在数据链路中分配资源, 144
- 流中的资源分配, 158
- 监视网络通信统计数据, 167
- 管理 EVS, 94
- 管理环, 145
- 配置虚拟网络组件, 21

MAC 地址, 在 VNIC 上, 14

MAC 客户机, 144

- 主, 147
- 分配环, 147
- 基于硬件的, 144
- 基于软件的, 144
- 配置, 147

N

内部虚拟网络, 13

NIC 环, 18, 18

- 参见 环分配
- 接收和传送, 144

O

Oracle Solaris 弹性虚拟交换机, 85, 87

- 参见 EVS

Oracle Solaris 内核区域, 99

Oracle VSI 管理器, 71, 77

oracle_v1 编码, 71

- 定义, 77

ORACLE_VSIMGR_V1, 77

P

配置

- VXLAN, 57

配置流, 158

- 命令, 158

配置网络记帐, 175

Q

迁移

- VF VNIC, 47

VNIC, 41

区域, 13, 94

临时创建 VNIC, 30

为虚拟网络重新配置, 28

使用 VXLAN, 55

分配 VXLAN, 62

指定 VNIC, 28

配置网络虚拟化, 26

S

删除 VNIC, 42

删除 VXLAN, 62

删除连接到区域的 VNIC, 44

上行链路端口

定义, 87

设置与 EVB 相关的数据链路属性的示例, 82

使用 LLDP 管理 VM 之间的通信, 76

使用 VDP 交换 VNIC 信息, 77

数据链路

EVB 属性, 80

分配 CPU, 18

分配 CPU 池, 18

分配带宽, 18

显示网络通信的历史统计数据, 176

资源控制属性, 19, 144

数据链路属性

cpu, 143

maxbw, 143

pool, 143

rxrings 和 txrings, 143

数据中心整合, 18

SCTP, 157

SLA

虚拟交换机和, 85

SR-IOV

为数据链路启用, 45

和 VNIC, 45

检查数据链路, 45

虚拟功能 见 VF

SSH 验证

设置, 105

T

弹性虚拟交换机, 85, 98

- 关联 IP 地址块和, 87
- 关联虚拟端口和, 87
- 创建, 114
- 创建 VNIC, 115
- 创建 VNIC anet 资源, 116
- 删除, 127
- 基于 VLAN 进行配置, 130
- 基于 VXLAN 为租户配置, 136
- 必需软件包, 112
- 显示, 114, 118
- 显示属性, 120
- 概述, 85
- 监视, 128
- 管理, 117
- 规划, 102
- 设置属性, 119
- 资源
 - IP 网络, 89
 - 虚拟端口, 89
- TCP, 157

- U**
- UDP, 157

- V**
- VDP 见 VSI 搜索和配置协议
- VDP 统计数据, 显示命令, 79
- VDP TLV, 77
- VF 信息
 - 显示, 48
- VF VNIC
 - 创建, 46
 - 显示, 48
 - 迁移, 47
- VLAN, 87, 88
 - 作为 VNIC, 24
 - 修改 VNIC 的 VLAN ID, 38
 - 在区域中临时创建, 31
- VNI, 52
- VNIC, 13, 13, 14
 - 与区域结合使用, 28
 - 专用虚拟网络, 31
 - 作为 VLAN, 21
 - 使用 SR-IOV, 45
- 修改 MAC 地址, 39
- 修改 VLAN ID, 38
- 创建, 21
- 创建 VNIC IP 接口, 23
- 删除, 42
- 在区域中临时创建, 30
- 在区域中配置, 29
- 带有 VLAN ID, 24
- 指定 CPU 池资源, 155
- 指定给区域, 28
- 指定静态 IP 地址, 23
- 显示信息, 34
- 显示多个 MAC 地址, 35
- 更改底层链路, 41
- 管理, 34
- 系统创建的, 22
- 设置属性, 144
- 迁移, 41
- 配置, 22, 26
- VNIC anet 资源, 94, 97
- VPort
 - 删除, 126
 - 显示, 125
 - 显示属性, 124
 - 添加到弹性虚拟交换机, 114
 - 管理, 117, 123
 - 设置属性, 123
 - 重置, 126
- VPort 属性, 89
 - IP 地址, 89
 - MAC 地址, 89
 - SLA
 - 优先级, 89
 - 最大带宽, 89
 - 服务类, 89
- VSI 见 虚拟站实例
- VSI 版本, 77
- VSI 标识符, 77
- VSI 管理器, 77
 - oracle_v1, 77
- VSI 管理器 ID
 - ORACLE_VSIMGR_V1, 77
- VSI 类型 ID, 77
- VSI 配置文件, 77
- VSI 搜索和配置协议 (VSI discovery and configuration protocol, VDP)

- VDP TLV, 77
- 以太网链路的 VDP 状态, 79
- 显示 VDP 统计数据, 79
- VXLAN, 17, 51, 87, 88, 98
 - anet 资源, 55
 - VNIC, 55
 - VXLAN 结束点, 52
 - VXLAN 网段, 55
 - 下行链路, 62
 - 与区域结合使用, 55
 - 优点, 52
 - 分配给区域, 62
 - 创建 VXLAN 的示例, 60
 - 删除, 62
 - 命名约定, 52
 - 将 VXLAN 分配给区域的 anet 的示例, 63
 - 拓扑, 53
 - 显示, 61
 - 概述, 51
 - 要求, 57
 - 规划配置, 57
 - 配置 VXLAN, 57
 - 配置区域, 57
- VXLAN 网段 ID, 52, 57 见 VNI

W

- 外部虚拟网络, 13
- 网络记帐, 175, 175
- 网络通道, 18
- 网络通信统计数据
 - 显示数据链路, 170
 - 显示流, 172
 - 显示网桥, 172
 - 显示网络设备, 168
 - 显示链路, 167
 - 显示链路聚合, 171
- 网络统计数据, 167
 - 监视网络使用情况, 165
 - 通信历史信息, 176
- 网络虚拟化, 12
 - etherstub, 14
 - 区域, 15
 - 和网络资源管理, 12
 - 数据中心整合, 18
 - 用于实现的 dladm 子命令, 144

- 虚拟交换机, 14
- 虚拟网络, 12
- 网络资源
 - CPU, 18
 - CPU 池, 18
 - NIC 环, 18
 - 优先级, 18
 - 带宽, 18
- 网络资源管理, 12, 18, 143
 - CPU, 152
 - CPU 池, 152
 - NIC 环, 144
 - 用于实现的 dladm 子命令, 144
 - 益处, 19
 - 通过使用数据链路属性, 18, 19
 - 通过使用流, 18, 19
 - 通过设置数据链路和流属性, 160
- 伪以太网 NIC 见 etherstub
- 物理链路状态
 - 显示, 36

X

- 系统创建的 VNIC, 22
 - 显示, 35
- 显示
 - EVS 控制器属性, 110
 - IPnet, 122
 - VDP 和 ECP 状态和统计数据, 79
 - VPort, 125
 - 与 EVB 相关的数据链路属性, 83
 - 弹性虚拟交换机信息, 112, 114, 118
 - 物理链路状态, 36
 - 虚拟链路状态, 36
- 显示 ECP 状态和统计数据的示例, 80
- 显示 VF, 48
- 显示 VNIC 上与 EVB 相关的属性的示例, 83
- 显示 VXLAN, 61
- 显示网络通信的历史统计数据
 - 在数据链路上, 176
 - 在流上, 176
- 显示物理链路上与 EVB 相关的数据链路属性的示例, 83
- 向数据链路分配 CPU, 156
- 协议
 - DCBX, 68

- ECP, 77
- VDP, 77
- 虚拟 NIC 见 VNIC
- 虚拟端口, 85 见 VPort
- 虚拟交换机, 13, 14, 85, 85
- 虚拟可扩展局域网 见 VXLAN
- 虚拟链路状态
 - 显示, 36
- 虚拟网络, 13, 15, 25, 52
 - 参见 VNIC
 - etherstub, 13
 - VNIC, 13
 - 专用, 13
 - 专用 IP 类型区域, 29
 - 区域, 13, 25
 - 外部和内部, 13
 - 实现, 18
 - 构建, 25
 - 组件, 13
 - 虚拟交换机, 13
 - 配置 etherstub, 22
 - 配置 VNIC, 22
 - 配置区域, 26
 - 重新配置区域, 28
- 虚拟网络接口卡 见 VNIC
- 虚拟网络组件
 - 配置, 21

Y

- 硬件环, 144
- 用于配置流的属性
 - DS 字段, 157
 - 传输协议, 157
 - 传输协议和应用程序端口号, 157
 - 传输协议和应用程序端口号及 IP 地址, 157
 - 本地 IP 地址, 157
 - 远程 IP 地址, 157
- 优先级
 - 数据链路, 18
 - 流, 18, 158
- 远程 IP 地址, 157
- 远程应用程序端口号, 157
- 允许
 - VM 通过外部交换机通信, 72

Z

- 在区域中临时创建 VNIC, 31
- 在区域中临时创建 VNIC、VLAN 和 IPoIB 分区, 31
- 主客户机, 144, 147
- 专用虚拟网络, 13, 14
 - 配置 etherstub, 31
- 自动生成的 VXLAN 数据链路, 98
- 子网, 89
- 租户
 - 定义, 90
- zoneadm 命令, 26
- zonecfg 命令, 26, 94, 155, 163

