Strategies for Network Administration in Oracle® Solaris 11.2

Part No: E37473-02
September 2014

Contents

Using This Documentation

- **Overview** – Provides information about networking strategies and describes how to use networking features to administer network configuration in the Oracle Solaris operating system (OS).
- **Audience** – System administrators.
- **Required knowledge** – Basic understanding of network administration concepts and practices.

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://www.oracle.com/pls/topic/lookup?ctx=E36784.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Summary of Oracle Solaris Network Administration

This chapter provides a summary of network administration in the Oracle Solaris release, including specific information about the network virtualization features that are supported.

For examples of network configuration scenarios, including basic network configuration of a host client system, as well as network virtualization use cases, see Chapter 2, "Network Configuration Scenarios".

For a shortcut to commonly used networking commands, see Chapter 3, "Oracle Solaris Network Administration Command Cheatsheet".

This chapter contains the following topics:

Highlights of Network Administration in Oracle Solaris

Users rely on various networking technologies to communicate, share, store, and process information. One of the primary goals of network administration is to establish and maintain reliable, secure, and efficient data communications on systems that are running the Oracle Solaris release. See "Basic Network Configuration in Oracle Solaris" on page 8.

Beyond the basic configuration that is required to connect client systems to the network, Oracle Solaris also supports several advanced networking technologies, including features that provide support for following functional areas:

- High availability
- Network security

- Network storage
- Network virtualization
- Observability, monitoring, and debugging
- Performance and efficiency
- Resource management

Most of these features are designed to address the complexities of modernized network environments by enabling you to use a more modular and layered approach to administering the various aspects of network configuration. For more information, see “Key Oracle Solaris Network Administration Features” on page 9 and “Network Administration by Functional Area” on page 14.

Basic Network Configuration in Oracle Solaris

Basic network configuration of a client system evolves in two stages: assembling the hardware and then configuring the daemons, files, and services that implement the network protocol stack. For more information about how the various networking components are configured within the network protocol stack, see “Network Administration Within the Oracle Solaris Network Protocol Stack” on page 12.

For an example of the information that is described in this section, see “Basic Network Configuration Scenario” on page 27.

The basic network configuration process typically involves the following tasks:

- You first customize the physical datalinks on the system. Each datalink represents a link object in the second layer (L2) of the Open Systems Interconnection (OSI) model. In this release, generic names are automatically assigned to datalinks by using the `net0`, `net1`, `net`*N* naming convention. The name that is assigned to each datalink depends on the total number of network devices that are on that system. For more information, see Chapter 2, “Administering Datalink Configuration in Oracle Solaris,” in “Configuring and Administering Network Components in Oracle Solaris 11.2 ”.
- After you customize the datalinks on a system, you then configure IP interfaces and addresses over each datalink. This configuration takes place at the network layer (L3) of the OSI model. You obtain unique IP addresses to communicate to public networks on the Internet. See Chapter 3, “Configuring and Administering IP Interfaces and Addresses in Oracle Solaris,” in “Configuring and Administering Network Components in Oracle Solaris 11.2 ”.

  Oracle Solaris supports both IPv4 and IPv6 configuration. You can choose whether to deploy a purely IPv4 network, an IPv6 network, or a network that uses a combination of both types of IP addresses. Deploying an IPv4 or IPv6 network requires some advanced planning. For more information about deploying a physical network in an organized and cost-effective manner, see “Planning for Network Deployment in Oracle Solaris 11.2 ”.
- Naming services and other system-wide network settings are fundamental to any computing network. These services perform lookups of stored information such as

host names and addresses, user names, passwords, access permissions, and so on. The information is made available so that users can log in to their host, access resources, and be granted permissions. Naming service information is centralized in the form of files, maps, and database files to make network administration more manageable. In this release, naming services are managed through the Service Management Facility (SMF). For more information about configuring system-wide network settings on an Oracle Solaris client, see Chapter 4, "Administering Naming and Directory Services on an Oracle Solaris Client," in "Configuring and Administering Network Components in Oracle Solaris 11.2 ".

- Network administration might also involve configuring systems that perform specific functions within your network, for example, routers, IP tunnels, and so on. For additional information, see "Configuring an Oracle Solaris 11.2 System as a Router or a Load Balancer " and "Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.2 ".

Before beginning the task of configuring a client system on the network, see "Information That Is Required to Configure Client Systems on the Network" in "Configuring and Administering Network Components in Oracle Solaris 11.2 ".

Key Oracle Solaris Network Administration Features

Oracle Solaris supports several networking features that you can use for different purposes. The following are some of the key features that are supported in this release. This list is not exhaustive:

- **Aggregation** – Is an L2 entity that is used to ensure that a system has continuous access to the network. Link aggregations increase the availability and reliability of network connectivity by enabling you to pool multiple datalink resources that you administer as a single unit. See Chapter 2, "Configuring High Availability by Using Link Aggregations," in "Managing Network Datalinks in Oracle Solaris 11.2 ".

  The following types of aggregations are supported:

  - **Datalink Multipathing (DLMP)** – Is a type of link aggregation that supports multiple switches and provides continuous connectivity to its datalinks. When a switch fails, the aggregation continues to provide connectivity to its datalinks by using the other switches. This type of link aggregation does not require switch configuration. Using DLMP aggregation can help overcome some of the disadvantages of using trunk aggregation. See "Datalink Multipathing Aggregations" in "Managing Network Datalinks in Oracle Solaris 11.2 ".
  - **Trunk aggregation** – Is a link aggregation mode that is based on the IEEE 802.3ad standard and works by enabling multiple flows of traffic to be spread across a set of aggregated ports. IEEE 802.3ad requires switch configuration and switch-vendor proprietary extensions in order to work across multiple switches. See "Trunk Aggregations" in "Managing Network Datalinks in Oracle Solaris 11.2 ".

- **Bridging** – Is an L2 technology that connects multiple datalinks on a network into a single network. For bridging, Oracle Solaris supports the Spanning Tree Protocol (STP) and TRansparent Interconnection of Lots of Links (TRILL) protocols. See Chapter 4, "Administering Bridging Features," in "Managing Network Datalinks in Oracle Solaris 11.2 ".
- **Edge Virtual Bridging (EVB)** – Is an L2 technology that enables hosts to exchange virtual link information with an external switch. EVB offloads the enforcement of traffic service level agreements (SLAs) to the switch. See Chapter 4, "Administering Server-Network Edge Virtualization by Using Edge Virtual Bridging," in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".
- **Data Center Bridging (DCB)** – Is an L2 technology that is used to manage the bandwidth, relative priority, and flow control of multiple traffic types that share the same network link, for example, when sharing a datalink between networking and storage protocols. See Chapter 6, "Managing Converged Networks by Using Data Center Bridging," in "Managing Network Datalinks in Oracle Solaris 11.2 ".
- **Elastic Virtual Switch (EVS)** – Is an L2 technology that expands network virtualization capabilities by enabling you to manage virtual switches across multiple hosts. With the Oracle Solaris EVS feature, you can deploy virtual networks that span multiple hosts within either a multi-tenant cloud environment or datacenter. See Chapter 6, "Administering Elastic Virtual Switches," in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".
- **Etherstub –** Is a pseudo Ethernet NIC that is configured at the datalink layer (L2) of the Oracle Solaris network protocol stack. You can create virtual interface cards (VNICs) over etherstubs instead of physical links for the purpose of constructing a private virtual network that is isolated from other virtual networks on the system, as well as from the external network. See "How to Configure VNICs and Etherstubs" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".
- **Flows –** Are a subset of packets that are identified by common attributes. These attributes consist of packet header information such as IP addresses, protocol type, and transport port numbers. You can observe flows individually, as well as assign flows their own SLAs, for example, bandwidth control and priority. You administer flows at the L2, L3, and L4 layers of the Oracle Solaris network protocol stack. For more information, see "Features for Managing Network Resources in Oracle Solaris" on page 23.
- **Integrated Load Balancer (ILB)** – Is an L3 and L4 technology that enables a system to spread the load of network processing amongst available resources. ILB can be used to improve reliability and scalability, and to minimize the response time of network services. Load balancing involves using multiple systems to deal with high demands of a network by balancing the load between multiple systems. Support for ILB in Oracle Solaris includes stateless Direct Server Return (DSR) and Network Address Translation (NAT) modes of operation for IPv4 and IPv6, as well as server monitoring capabilities through health checks. See "Features of ILB" in "Configuring an Oracle Solaris 11.2 System as a Router or a Load Balancer ".
- **IP network multipathing (IPMP)** – Is an L3 technology that ensures that a system has continuous access to the network. With IPMP, you configure multiple IP interfaces into an *IPMP group*. The IPMP group functions like an IP interface with data addresses for

sending or receiving network traffic. If an underlying interface in the group fails, the data addresses are redistributed amongst the remaining underlying active interfaces in the group.

The IPMP model and administrative interface has undergone some changes in Oracle Solaris 11. To familiarize yourself with the new model, see "What's New in IPMP" in "Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.2 ".

Link aggregations work similarly to IPMP for improving network performance and availability, but at the datalink layer (L2). Aggregations are recommended when combining features for high availability in a virtualized environment. For a comparative analysis, see Appendix A, "Link Aggregations and IPMP: Feature Comparison," in "Managing Network Datalinks in Oracle Solaris 11.2 ".

- **IP tunnel** – Is an L3 technology that provides a means for transporting data packets between domains when the protocol in those domains is not supported by intermediary networks. See Chapter 4, "About IP Tunnel Administration," in "Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.2 ".
- **Link Layer Discovery Protocol (LLDP)** – Is an L2 technology that is used by systems in a local area network (LAN) to exchange configuration and management information with each other. LLDP enables a system to advertise connectivity and management information to other systems on the network. See Chapter 5, "Exchanging Network Connectivity Information With Link Layer Discovery Protocol," in "Managing Network Datalinks in Oracle Solaris 11.2 ".
- **Virtual local area network (VLAN)** – Is an L2 technology that enables you to divide a LAN into subnetworks without having to add a physical network environment. A VLAN is a subdivision of a LAN at the datalink layer of the network protocol stack. For more information, see Chapter 3, "Configuring Virtual Networks by Using Virtual Local Area Networks," in "Managing Network Datalinks in Oracle Solaris 11.2 ".
- **Virtual eXtensible area network (VXLAN)** – Is an L2 and L3 technology that works by overlaying a datalink (L2) network on top of an IP (L3) network. VXLANs address the 4K limitation that is imposed when using VLANs. Typically, VXLANs are used in a cloud infrastructure to isolate multiple virtual networks. You can manage VXLANs by using the EVS feature. For more information, see Chapter 3, "Configuring Virtual Networks by Using Virtual Extensible Local Area Networks," in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".
- **Virtual network interface card (VNIC)** – Is an L2 entity or virtual network device that behaves just like a physical NIC when configured. You configure a VNIC over an underlying datalink to share it between multiple Oracle Solaris zones or VMs. See "Configuring the Components of a Virtual Network" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

  In this release, you can also manage network devices that support single `root` I/O virtualization (SR-IOV). For more details, see "Using Single Root I/O Virtualization With VNICs" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".
- **Virtual Router Redundancy Protocol (VRRP)** – Is an L3 technology that provides high availability of IP addresses, such as those that are used for routers and load balancers.

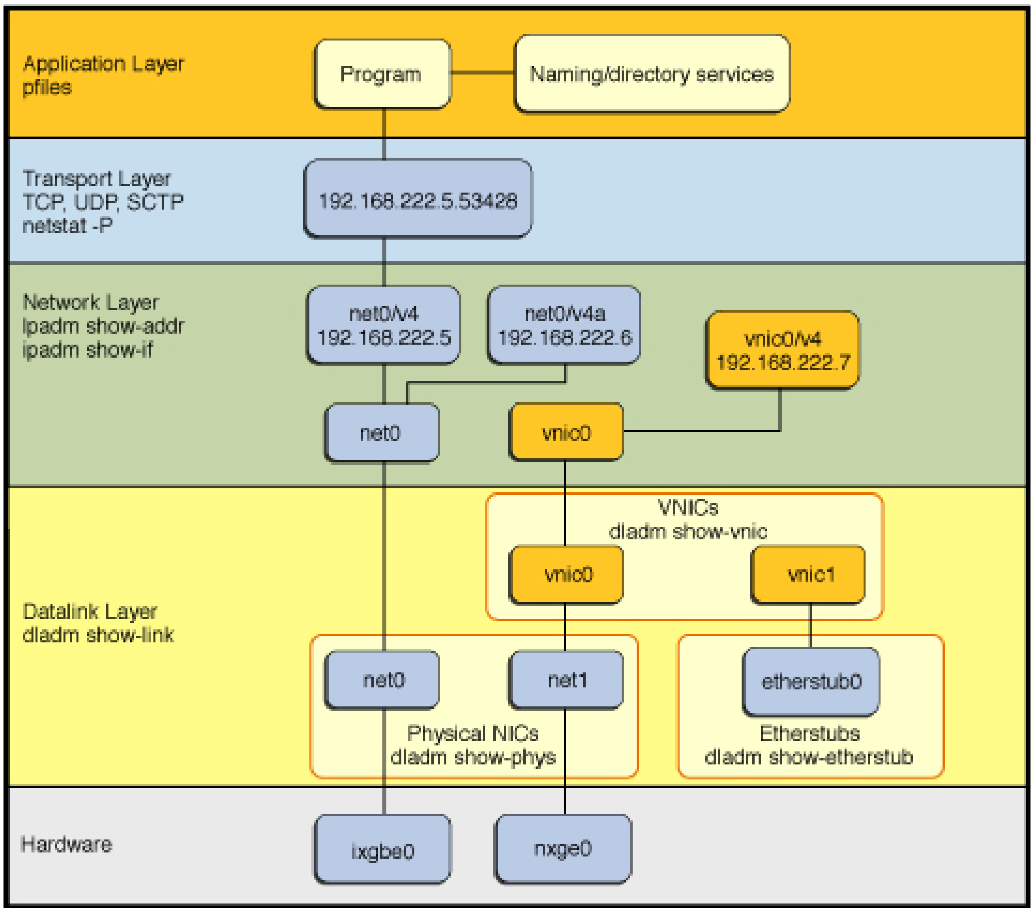
Oracle Solaris supports both L2 and L3 VRRP. L3 VRRP removes the need to configure unique VRRP virtual MAC addresses for VRRP routers, thereby providing better support for VRRP over IPMP, InfiniBand interfaces, and zones. For more information, see Chapter 3, "Using Virtual Router Redundancy Protocol," in "Configuring an Oracle Solaris 11.2 System as a Router or a Load Balancer ".

- **Virtual switch** – Is an L2 technology that simulates the capabilities of a physical network switch. A virtual switch is implicitly created whenever you create a VNIC on top of an underlying datalink. Virtual switches provide a method for virtual machines and zones to transfer packets. You can manage virtual switches by using the EVS feature. For more information, see "Components of a Virtual Network" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

Network Administration Within the Oracle Solaris Network Protocol Stack

The following figure shows the layers of the Oracle Solaris network protocol stack and where within the stack both physical and virtual interfaces are administered. This information can be helpful when planning which networking strategy or strategies to deploy at your site. Knowing which layer of the network protocol stack a particular feature is configured is also helpful when troubleshooting network configuration issues, detecting network connectivity problems, and diagnosing performance issues, such as packet loss. The information in Table 1-1 provides additional details about where each feature is administered within the Oracle Solaris network protocol stack.

FIGURE 1-1 Physical and Virtual Network Administration Within the Network Protocol Stack



The following table further describes which layer of the Oracle Solaris network protocol stack each networking feature is administered. Note that some features are administered in more than one layer of the stack.

Note - Only those layers of the network protocol stack that pertain to the various network administration features described in this document are shown.

TABLE 1-1 Networking Features by Network Protocol Stack Layer

Network Protocol Stack Layer	Feature or Technology
Transport (L4)	■ Firewalls ■ Flows ■ Pluggable congestion control ■ Socket filtering
Protocol or network (L3)	■ DHCP ■ Flows ■ IP interfaces and IP addresses ■ IP tunnels ■ IPMP ■ ILB ■ Routing ■ VNIs ■ VRRP ■ VXLANs

Datalink (L2)	■ Aggregations (DLMP and trunking)
	■ EVB
	■ Flows
	■ LLDP
	■ Physical datalinks
	■ Network virtualization features:
	  ■ DCB
	  ■ Etherstubs
	  ■ EVS
	  ■ Virtual switches
	  ■ VLANs
	  ■ VNICs
	  ■ VXLANs

Network Administration by Functional Area

Oracle Solaris network administration features are designed to meet specific networking needs by providing support in the following functional areas: high availability, network virtualization, performance, resource management, security, and storage. Knowing which functional area a particular feature supports is helpful for evaluating which networking strategy or strategies to implement at your site.

The following table describes the various network administration features that are supported in Oracle Solaris according to functional area. Information about the administrative interface that is used to administer the feature and at which layer of the network protocol stack the feature is administered, is also provided.

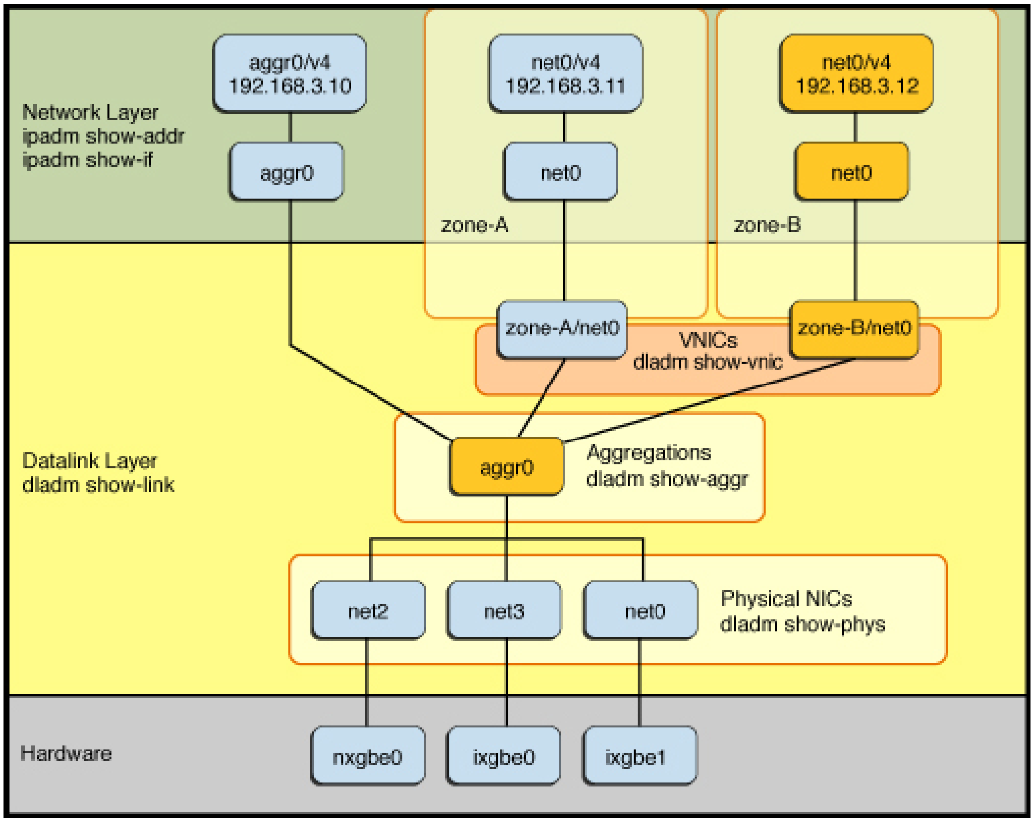
TABLE 1-2 Networking Features by Functional Area

Feature	Functional Area	Administrative Interface	Network Protocol Stack Layer
Aggregations (DLMP and trunking)	High availability	`dladm` (`create-aggr`, `delete-aggr`, `modify-aggr`, `add-aggr`, `remove-aggr`)	L2
Bridging protocols: ■ STP ■ TRILL	High availability, network virtualization	`dladm` (`create-bridge`, `delete-bridge`, `modify-bridge`, `add-bridge`, `remove-bridge`, `show-bridge`)	L2
DCB	Network storage, performance	`lldpadm`, `dladm`	L2
Etherstubs	Network virtualization	`dladm` (`create-etherstub`, `delete-etherstub`, `show-etherstub`)	L2
EVB	Network virtualization	`dladm`	L2
EVS	Network virtualization	`evsadm`, `evsstat`, `dladm`	L2, L3
Firewalls	Security	Packet filtering with `ipf` and `ipnat`	L3, L4
Flows	Observability, resource management, security	`flowadm`, `flowstat`	L2, L3, L4
ILB	Performance	`ilbadm` (`create-servergroup`, `add-server`, `delete-servergroup`, `enable-server`, `disable-server`, `show-server`, `show-servergroup`, `remove-server`)	L3
IPMP	High availability	`ipadm` (`create-ipmp` *interface*, `delete-ipmp` *interface*, `add-ipmp` *interface*, `remove-ipmp` *interface*	L3
IP tunnels	IP connectivity	`dladm` (`create-iptun`, `modify-iptun`, `delete-iptun`, `show-iptun`); `ipadm` (to create the IP address over the tunnel)	L2, L3
LLDP	Observability, network storage, network virtualization	`lldpadm`	L2
Pluggable congestion control	Performance	`ipadm set-prop` *property*	L4
Routing	IP connectivity	`route` (`route -p display`; `netstat`); `routeadm`	L3
Socket filtering	Security	`soconfig` (`-F`)	L4

Feature	Functional Area	Administrative Interface	Network Protocol Stack Layer
VLANs	Network virtualization	`dladm` (`create-vlan`, `modify-vlan`, `delete-vlan`, `show-vlan`)	L2
VNIs	IP connectivity	`ipadm` (`create-vni`, `delete-vni`)	L3
VNICs	Network virtualization	`dladm` (`create-vnic`, `modify-vnic` `delete-vnic`, `show-vnic`)	L2
VRRP	High availability	`dladm`, `vrrpadm`	L3
VXLANs	Network virtualization	`dladm` (`create-vxlan`, `show-vxlan`, `delete-vxlan`)	L2, L3

In many cases, you can obtain optimal results by using a combination of networking features. For example, the following figure shows how you might combine multiple networking features for high availability.

FIGURE 1-2 Combining the Use of Aggregations With VNICs



In the figure, multiple physical datalinks (`net0`, `net2`, and `net3`) are combined into a single link aggregation (`aggr0`). The aggregation datalink is then directly configured from IP in the global zone through the `aggr0` and `aggr0` IP interface and IP address, respectively. For another example, see "Combining Aggregations With VNICs for High Availability" on page 31.

You can also virtualize the aggregation datalink by using it as the underlying link for the VNICs. In this figure, two VNICs are configured and then assigned to two non-global zones. This particular configuration makes the VNICs highly available because any failures of the underlying physical NICs that occur are automatically handled by the link aggregation layer and are transparent to the zones.

Summary of Network Virtualization in Oracle Solaris

With server virtualization becoming more mainstream in the IT industry, the focus is shifting to a deployment model that uses network virtualization to support the sharing of network traffic amongst multiple virtual machines (VMs) or zones. Along with a rise in the adoption of cloud architectures that rely upon virtualization for deploying workloads, network virtualization is playing an even more critical role in the overall network administration strategy in Oracle Solaris.

Virtual environments require a high degree of availability, insulation, performance, and separation. Oracle Solaris provides several features that meet these requirements. Additionally, Oracle Solaris network virtualization features are tightly integrated with other Oracle Solaris features (subsystems). For example, when configuring a zones environment, you can create VNICs (`anets`) that are automatically configured when the zone boots. For information about working with Oracle Solaris zones, see "Introduction to Oracle Solaris 11.2 Virtualization Environments ".

Network virtualization is also tightly integrated with Oracle Solaris resource management features, which are used to limit the CPUs in a zones environment. For more information about network virtualization and resource management features in Oracle Solaris, see "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

For information about Oracle VM, including Oracle VM Server for x86, Oracle VM Server for SPARC (previously called Sun Logical Domains, or LDoms), and Oracle VM Manager, see the documentation at http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html.

Oracle also provides the Oracle Enterprise Manager Ops Center for managing some aspects of network virtualization, for example, the ability to create virtual private networks inside a virtual datacenter. For more information about Oracle Enterprise Manager Ops Center, refer to the *Certified Systems Matrix* document at http:/www.oracle.com/pls/topic/lookup?ctx=oc122.

For more information and examples, see the scenarios that are described in Chapter 2, "Network Configuration Scenarios".

Network Virtualization Building Blocks

Network virtualization in Oracle Solaris includes the following key building blocks:

- **VNICs**

  When a datalink such as a physical NIC or a link aggregation needs to be shared by multiple VMs or zones, you can carve it up into *virtual NICs* or VNICs. These VNICs appear on the system as any other NIC and are administered exactly the same way as a physical NIC. Each VNIC has its own MAC address that you can configure with additional attributes, such as a *VLAN ID*, thus enabling the VNIC to be easily integrated into an existing network infrastructure. For higher availability, you can also create VNICs on top of link aggregations, which you can then assign individual bandwidth limits, thereby enabling them to consume just their assigned share of bandwidth. VNICs have a rich set of configurable features. For more information, see "Building Virtual Networks" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

- **Virtual switching**

  The Oracle Solaris virtual network stack includes built-in virtual switching capabilities that simulate the capabilities of a physical network switch. You can use virtual switches within a single machine to enable zones and VMs to communicate with each other. Virtual switches are automatically instantiated when multiple VNICs are created on top the same datalink. In addition to being able to create VNICs on top of physical NICs or aggregations, you can also create virtual switches on top of an *etherstub*. This capability enables you to create fully virtualized networks that are independent of the physical hardware. For more information, see "Configuring the Components of a Virtual Network" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

- **Elastic Virtual Switch (EVS) feature of Oracle Solaris**

  The EVS feature is an L2 technology that expands network virtualization capabilities by enabling you to directly manage virtual switches. You can create EVS switches to deploy multiple virtual networks that span multiple hosts, within either a multi-tenant cloud environment or a datacenter. You can also optionally configure an EVS switch with virtual ports, IP subnets, and Service Level Agreements (SLAs). Additionally, you can connect any Oracle Solaris VNIC to an EVS switch or a virtual port. Such VNICs automatically inherit their network configuration from EVS. This capability enables you to more cleanly separate the network configuration from a zone or VM configuration.

  You manage and observe EVS switches through a central *controller*. The elastic virtual switches are then automatically deployed on the various hosts, as needed. Hence, the term *elastic* is used to describe these switches. The EVS architecture is tightly integrated with various other network virtualization features, including the VXLAN feature. See Chapter 3, "Configuring Virtual Networks by Using Virtual Extensible Local Area Networks," in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".Together, you can use these two features to create a large number of virtual networks. Also, because EVS switches are transport-agnostic, you can use them with other types of network fabrics, such as traditional VLANs.

  EVS switches are also supported in a zones environment. An `anet` VNIC resource can connect to an EVS switch by using the appropriate `zonecfg` properties. See "Creating and Using Oracle Solaris Zones " and the `zonecfg`(1M) man page for more information.

  For more information about the EVS feature, see Chapter 6, "Administering Elastic Virtual Switches," in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

  The EVS feature introduces new administrative commands. For more information, see the `evsadm`(1M) and `evsstat`(1M) man pages. See also the `dladm`(1M) man page.

Network Virtualization Strategies

You can deploy Oracle Solaris network virtualization features for the following purposes:

- **Workload consolidation**

  It is a common practice in modern datacenters to consolidate multiple workloads onto a single machine. This type of workload consolidation is typically accomplished by using virtualization on multiple VMs or zones or by using a combination of both methods. To provide network access to these entities, Oracle Solaris network virtualization features provide a way for the physical NICs that are on a system to be virtualized into multiple VNICs. By virtualizing the physical NICs, you eliminate the need to have separate physical NICs for each VM or zone. The VM or zone shares the physical NIC. As is the case for other virtualized resources, it is important to control the share of the network resources to which each virtual machine is entitled. To accomplish this task, you can configure bandwidth limits on the individual VNICs. Using resource control along with VNICs can further improve the use of resources in a multiple virtual network stack.

- **Private virtual networks**

  You can also use network virtualization features to build private virtual networks for the following purposes:

  - **Security** – Create a private virtual network behind a virtual firewall to better insulate virtual machines from the physical network and also to better insulate the physical network from the virtual network.
  - **Testing and simulation** – Create a private virtual *network within a box* to test different features or simulate a feature's behavior under a given network load before actually implementing that new feature or network configuration.
  - **Network consolidation** – Consolidate multiple hosts, network functions, and various network devices such as routers, firewalls, load balancers, and so on, within a box.

- **Cloud networking**

  A *cloud architecture* is a network administration approach that uses a *utility computing model* to deploy workloads. In this administrative model, multiple tenants share the same cloud and therefore must be isolated from one another. A cloud architecture is highly dynamic.

  Oracle Solaris provides several network virtualization features that are ideal for this type of environment. For example, you could use the EVS feature to build virtual network topologies that span multiple hosts while providing a single point of control and observability.

  With EVS, a cloud administrator can easily provision, control, and observe a per-tenant, virtual network. This type of configuration includes the capability for satisfying the most demanding requirements for agility and security in modern cloud environments. For more information about how you would set this type of scenario up, see “Setting Up an EVS Virtual Tenant Network” on page 33.

  For background information, see Chapter 5, “About Elastic Virtual Switches,” in “Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ”.

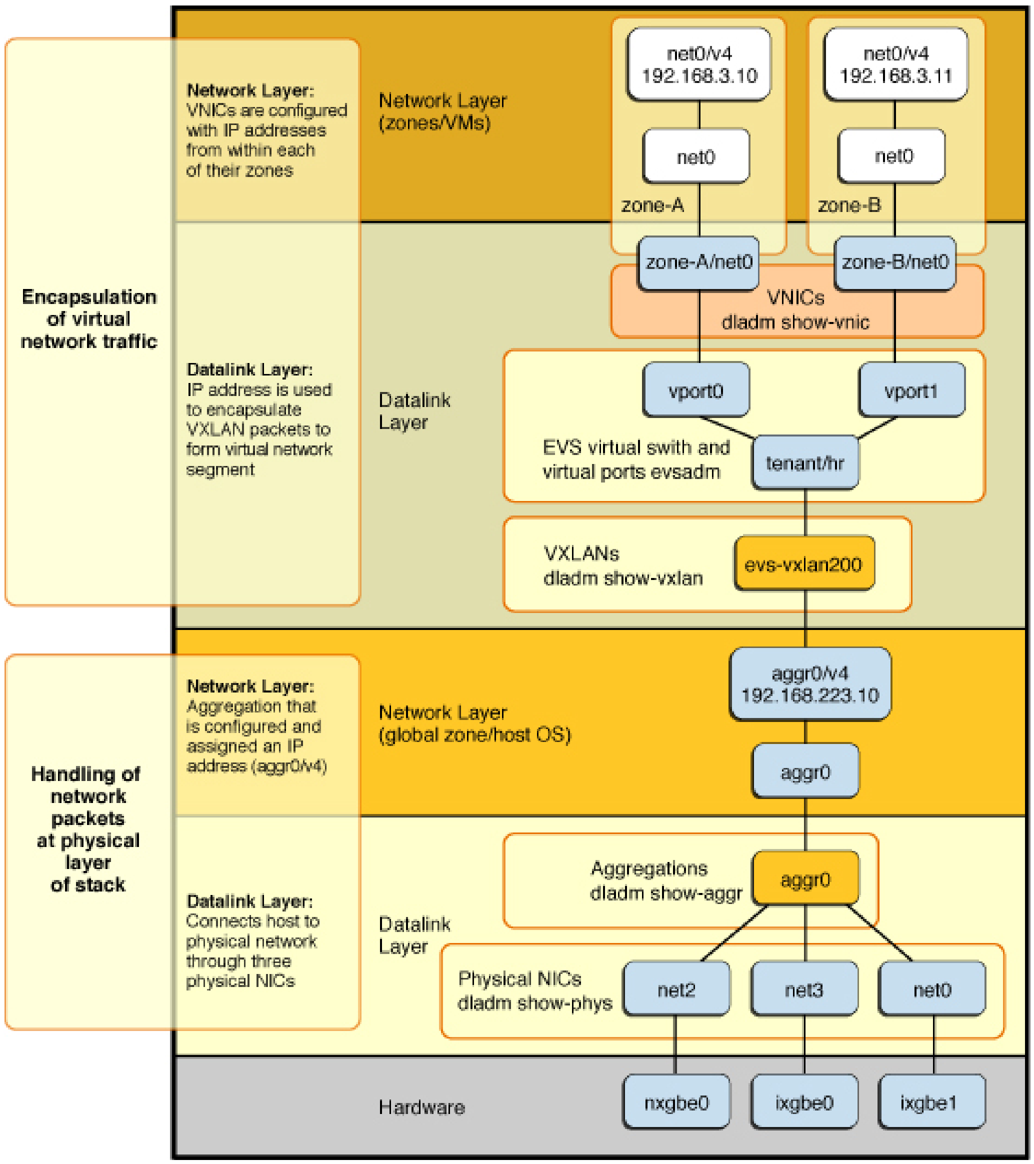
Creating a Highly Available Virtual Network Stack for Cloud Environments

The following figure shows how you might combine multiple network virtualization features, for example, link aggregations, VNICs, VXLANs, and an EVS switch, to provide a highly available and integrated virtual network stack for cloud environments.

In this figure the datalink and network layers of the network protocol stack appear twice. This layering is the result of using VXLANs which provide virtual network segments that are encapsulated within IP packets. The datalink and network layers therefore appear twice in the resulting network stack: once to show the handling of packets at the physical layer and then again to show the encapsulated virtual network traffic within these layers of the stack.

The datalink layer that is shown at the lower level of the figure (just above the hardware layer) is used to connect the host to the physical network through three physical NICs, which are aggregated to provide high availability. The resulting aggregation is configured at the network layer and assigned an IP address (`aggr0/v4`). The same IP address is then used to encapsulate VXLAN packets which form a virtual network segment. In Oracle Solaris, VXLANs are configured through datalinks and then used through VNICs. These VNICs are then configured with IP addresses from within their zones, as depicted in the datalink and network layers that are shown in the top portion of the figure.

FIGURE 1-3 Combining Aggregations With VXLANs, VNICs, and an EVS Switch



The figure represents the following configuration:

1. Starting from the hardware layer, multiple physical NICs (`net0`, `net2`, and `net3`) are aggregated to form a highly available link aggregation called `aggr0`.

2. The aggregation is then configured with an IP address, `aggr0/v4` (`192.168.223.10`).
3. An EVS virtual switch `tenant/hr` is created on top of the IP interface `aggr0`. In this figure, EVS is configured to use a VXLAN.

   The new `vxlan0` datalink is connected to a virtual L2 network that overlays the IP network.
4. Assuming that EVS assigned the virtual switch a VXLAN ID of 200, EVS automatically creates a VXLAN datalink called `evs-vxlan200`, which is associated with the `tenant/hr` virtual switch.
5. The EVS switch has two virtual ports (`vport0` and `vport1`), which are connected to two VNICs that are used by two zones. The VNICs appear in the zone as datalinks named `net0` and are visible from the global zone as `zone-A/net0` and `zone-B/net0`.

For some examples of how you would deploy these features, see Chapter 2, "Network Configuration Scenarios".

Features for Managing Network Resources in Oracle Solaris

Network resource management in Oracle Solaris consists of setting datalink properties that pertain specifically to how network resources are allocated. By setting these properties, you can determine how much of a given resource can be used for networking processes. For example, a link can be associated with a specific number of CPUs that are reserved exclusively for networking processes. Or, a link can be allotted a given bandwidth to process a specific type of network traffic.

Procedures for allocating resources apply to both virtual networks and traditional (physical) networks. For example, you can use the `dladm set-linkprop` command to set properties that are related to network resources. This same syntax is used for both physical and virtual datalinks.

Network resource management is comparable to creating dedicated lanes for traffic. When you combine different resources to cater to specific types of network packets, those resources form a network lane specifically for those network packets.

Use network resource management features to accomplish the following:

- Provision a network
- Establish service level agreements
- Bill clients
- Diagnose security problems

You can also use flows to manage network resources. A *flow* is a customized way of categorizing packets to further control how resources are used to process those packets. Network packets can be categorized according to an attribute. Packets that share an attribute constitute a flow and are labeled with a specific *flow name*. You can then assign specific resources to the flow.

The commands that you use to allocate network resources depend on whether you are working with datalinks or flows.

- For datalinks, use the `dladm` command with the appropriate subcommand, depending on whether you are setting the property during link creation or afterwards.
- For flows, use the `flowadm` command with the appropriate subcommand. Managing resources on flows parallels the methods for managing resources on datalinks.

  You can use the `flowadm add-flow` command to configure flows on a datalink based on a single attribute or a combination of attributes. Configuring flows based on a combination of attributes enables you to selectively organize network packets that are received from different ports, transport protocols, and IP addresses.

  The set of defined attributes that characterizes the flows constitutes the system's *flow control policy*.

For complete instructions, see Chapter 7, "Managing Network Resources," in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 " and the `dladm`(1M) and `flowadm`(1M) man pages.

Features for Managing Network Security in Oracle Solaris

Oracle Solaris provides several security features that enable you to protect and secure your network. The following table briefly describes several key network security features.

TABLE 1-3 Oracle Solaris Network Security Features

Features and Methods Used to Secure the Network	Description	For More Information
Link protection	The link protection mechanism provides protection from basic threats to a network, for example, IP, DHCP, and MAC spoofing, as well as L2 frame spoofing and Bridge Protocol Data Unit (BPDU) attacks.	Chapter 1, "Using Link Protection in Virtualized Environments," in "Securing the Network in Oracle Solaris 11.2 "
Network parameter tuning	Tuning network parameters ensures that the network is secure and prevents malicious attacks, for example, various types of denial-of-service (DoS) attacks.	Chapter 2, "Tuning Your Network," in "Securing the Network in Oracle Solaris 11.2 "
Secure Sockets Layer (SSL) protocol for web server communications	The SSL protocol encrypts and accelerates web server communications on your Oracle Solaris system. SSL provides confidentiality, message integrity, and endpoint authentication between two applications.	Chapter 3, "Web Servers and the Secure Sockets Layer Protocol," in "Securing the Network in Oracle Solaris 11.2 "

Features and Methods Used to Secure the Network	Description	For More Information
IP Filter feature of Oracle Solaris	Packet filtering provides basic protection against attacks on your network. The IP Filter feature of Oracle Solaris is a firewall that provides stateful packet filtering and network address translation (NAT). IP Filter also includes stateless packet filtering and the ability to create and manage address pools.	Chapter 4, "About IP Filter in Oracle Solaris," in "Securing the Network in Oracle Solaris 11.2 "
IP Security Architecture (IPsec)	IPsec provides cryptographic protection for IP datagrams in IPv4 and IPv6 network packets. IPsec includes several components that provide protection for IP packages by authenticating or encrypting the packets.	Chapter 7, "Configuring IPsec," in "Securing the Network in Oracle Solaris 11.2 "
Internet Key Exchange (IKE)	The IKE feature automates key management for IPsec. IKE easily scales to provide a secure channel for a large volume of traffic.	Chapter 9, "Configuring IKEv2," in "Securing the Network in Oracle Solaris 11.2 "

♦ ♦ ♦ CHAPTER 2

Network Configuration Scenarios

This chapter contains one basic network configuration scenario and three network virtualization scenarios. The basic network configuration scenario describes essential tasks for configuring an Oracle Solaris host client system on the network. The network virtualization scenarios describe networking strategies that combine multiple network virtualization features for high availability, optimal performance, resource management, and the deployment of workloads in a cloud environment.

For overview information about network administration, see Chapter 1, "About Network Administration in Oracle Solaris," in "Configuring and Administering Network Components in Oracle Solaris 11.2 ".

For more information about administering network virtualization features, see "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

This chapter contains the following topics:

Basic Network Configuration Scenario

To perform basic network configuration of an Oracle Solaris host client system, you must first customize the datalinks on the system. Then, you configure the IP interfaces and IP addresses, as well as add a persistent default route for the system. Additionally, you configure any system-wide network services such as naming and directory services. The following examples assume that you are using the *fixed mode* for network configuration. See Example 2-1.

Depending on your particular networking needs, you might not need to perform each of the following tasks to configure your network. Or, you might need to perform additional tasks that are not described in this scenario. For a quick reference to commonly used network administration commands, see Chapter 3, "Oracle Solaris Network Administration Command Cheatsheet".

This section contains the following topics:

- "Configure Datalinks, IP Interfaces, and IP Addresses" on page 28
- "Configure Naming Services Through SMF" on page 29
- "Set a System's Host Name" on page 31

Configure Datalinks, IP Interfaces, and IP Addresses

The following configuration tasks are described:

- Verify the current network configuration mode.
- Determine how network interface names on the system map to physical interfaces.
- Configure a static IP interface and address.
- Add a persistent default route.

EXAMPLE 2-1 Verifying the Active Network Mode

After installing Oracle Solaris, verify which configuration mode you are using as follows

```
# netadm list
TYPE        PROFILE        STATE
ncp         Automatic      disabled
ncp         DefaultFixed   online
loc         Automatic      offline
loc         NoNet          offline
loc         DefaultFixed   online
```

The previous output indicates that the system is using the fixed mode, which means you use the `dladm`, `ipadm`, and route commands to manage network configuration.

If the system-generated `Automatic` profile is online, enable the `DefaultFixed` profile as follows:

```
# netadm enable -p ncp DefaultFixed
```

EXAMPLE 2-2 Determining How Network Interface Names Map to Physical Interfaces

Prior to configuring an IP interface and static IP address for a system, determine how the network interface names on the system map to the physical interfaces. On a system with multiple physical networks, use the `dladm` command to obtain this information:

```
# dladm show-phys
LINK              MEDIA                STATE      SPEED  DUPLEX    DEVICE
net0              Ethernet             up         1000   full      e1000g0
```

```
net1              Ethernet             unknown    0      unknown    pcn0
```

EXAMPLE 2-3 Configuring a Static IP Address

You first create the IP interface and then you configure an IP address for the interface. You can associate multiple IP addresses with a single IP interface. In the following example, `ronj` is used for example purposes only.

```
# ipadm create-ip net0
# ipadm show-if
IFNAME     CLASS     STATE     ACTIVE     OVER
lo0        loopback  ok        yes        ---
net0       ip        down      no         ---
# ipadm create-addr -T static -a 10.163.198.20/24 net0/ronj
# ipadm show-if
IFNAME      CLASS     STATE     ACTIVE     OVER
lo0         loopback  ok        yes        ---
net0        ip        ok        yes        ---
# ipadm show-addr
ADDROBJ     TYPE      STATIC    ADDR
lo0/v4      static    ok        127.0.0.1/8
net0/ronj   static    ok        10.163.198.20/24
lo0/v6      static    ok        ::1/128
```

If your site implements IPv6 addressing, use the `addrconf` argument with the `-T` option to specify an automatically generated IPv6 address:

```
# ipadm create-ip net0
# ipadm create-addr -T addrconf net0/addr
```

If you need to obtain the IP address from a DHCP server, type the following commands:

```
# ipadm create-ip net0
# ipadm create-addr -T dhcp net0/addr
```

EXAMPLE 2-4 Adding a Persistent Default Route

After configuring an IP interface and address, add a persistent default route as follows:

```
# route -p add default 10.163.198.1
add net default: gateway 10.163.198.1
add persistent net default: gateway 10.163.198.1
```

For detailed instructions, see "Creating Persistent (Static) Routes" in "Configuring and Administering Network Components in Oracle Solaris 11.2 ".

Configure Naming Services Through SMF

Because the SMF repository is the primary repository for all naming services configuration in Oracle Solaris 11, the previous method of modifying a configuration file to configure naming services no longer works. If you make changes to any of these services, for example, `svc:/system/name-service/switch`, `svc:/network/dns/client`, or `svc:/system/name-service/cache`, you must enable and refresh the service for the changes to take effect.

Note - If no network configuration exists, naming services default to `files only` behavior, rather than `nis files`. Note also that the `svc:/system/name-service/cache` SMF service should be enabled at all times.

The following configuration tasks are described:

- Configure DNS.
- Set multiple DNS options.
- Set multiple NIS servers.

EXAMPLE 2-5 Configuring DNS Through SMF

The following example shows how to configure the Domain Name Service (DNS) by using SMF commands. The DNS configuration on a system provides the ability to look up IP addresses by host name and host names by IP address. As shown in this example, you can set DNS properties from the command line, or you can set the same properties interactively. See "Configuring a DNS Client" in "Configuring and Administering Network Components in Oracle Solaris 11.2 " for an example. After you set the various properties, you must enable and refresh the SMF service for the changes to take effect.

```
# svccfg -s dns/client setprop config/nameserver=net_address: 192.168.1.1
# svccfg -s dns/client setprop config/domain = astring: "myhost.org"
# svccfg -s name-service/switch setprop config/host = astring: "files dns"
# svcadm refresh name-service/switch
# svcadm refresh dns/client
```

EXAMPLE 2-6 Configuring Multiple DNS Options Through SMF

One network configuration task that you might need to perform is to set DNS options for a system. The following example shows how to set multiple `/etc/resolv.conf` options simultaneously.

```
# svccg
svc:> select /network/dns/client
svc:/network/dns/client> setprop config/options = "ndots:2 retrans:3 retry:1"
svc:/network/dns/client> listprop config/options
config/options astring     ndots:2 retrans:3 retry:1

# svcadm refresh dns/client
# grep options /etc/resolv.conf
options ndots:2 retrans:3 retry:1
```

EXAMPLE 2-7 Configuring Multiple NIS Servers Through SMF

The following example shows how to set multiple NIS servers simultaneously.

```
# svccfg -s nis/domain setprop config/ypservers = host: (1.2.3.4 5.6.7.8)
```
(Note the space between 1.2.3.4 and 5.6.7.8)

Set a System's Host Name

Note - The primary interface's TCP/IP host name is a distinct entity from the *system host name* that you set with the `hostname` command. Although not required by Oracle Solaris, the same name is normally used for both. Some network applications depend on this convention.

Permanently set a system's host name as follows:

```
# hostname name-of-host
```

Initially, the `hostname` value is stored in `config/nodename`, but this value is overridden if the system is configured by using DHCP, in which case, DHCP provides the `hostname` value. If the `hostname` command is used, then `hostname` is the value specified in the `config/nodename` file. If you set a system's identity by using the `hostname` command, this setting cannot be overridden by DHCP until you execute the `hostname` command with the `-D` option. The corresponding SMF properties and the associated SMF service are also automatically updated when you use the `hostname` command. See the `hostname(1)` man page.

Combining Aggregations With VNICs for High Availability

The following scenario describes how to combine a Datalink Multipathing (DLMP) aggregation with VNICs for high availability. Figure 1-2 graphically depicts this type of configuration.

The system that is used to create and configure the DLMP aggregation in the following example has a set of 10 Gigabit Ethernet NICs, as shown in the following output:

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

EXAMPLE 2-8 Configuring and Virtualizing DLMP Aggregations With VNICs

1. First, create the DLMP aggregation (`aggr0`) with probing enabled for the `net1` and `net2` interfaces, as shown in the following example:

```
# dladm create-aggr -l net1 -l net2 -m dlmp -p probe-ip=+ aggr0
```

Setting the `probe-ip` property enables probe-based detection failure with automatic selection of the source and target probing IP addresses. For details, see "Configuring Probe-Based Failure Detection for DLMP Aggregation" in "Managing Network Datalinks in Oracle Solaris 11.2 ".

Then, create an IP interface and address for the aggregation datalink as follows:

```
# ipadm create-ip aggr0
# ipadm create-addr -T dhcp aggr0
```

2. Virtualize the DLMP aggregation.

   You can then easily virtualize the aggregation by creating a VNIC over the aggregation datalink. For example, you would create a VNIC over `aggr0` as follows:

```
# dladm create-vnic -l aggr0 vnic0
```

   The newly created VNIC (`vnic0`) is now highly available. If one of the aggregated links (`net1` or `net2`) fails, the traffic for that VNIC automatically fails over to the remaining link and the operation is transparent to the VNIC.

   View information about the aggregation by using either of the following commands:

```
# dladm show-aggr
LINK            MODE  POLICY   ADDRPOLICY           LACPACTIVITY LACPTIMER
aggr0           dlmp  --       --                   --           --


# dlstat show-aggr -x
LINK       PORT           SPEED  DUPLEX   STATE         ADDRESS           PORTSTATE
aggr0      --             1000Mb  full    up            0:14:4f:fa:ea:42  --
           net1           1000Mb  full    up            0:14:4f:fa:ea:42  attached
           net2           1000Mb  full    up            0:14:4f:f9:c:d    attached
```

EXAMPLE 2-9 Specifying an Aggregation Datalink as the Lower Link of a Zone's `anet` Resource

Alternatively, you can virtualize an aggregation for high availability by specifying the aggregation datalink as the lower link of an Oracle Solaris zone's `anet` resource, as shown in the following example. Or, you can specify an aggregation datalink as the uplink of an EVS node. See "Setting Up an EVS Virtual Tenant Network" on page 33 for an example of this type of configuration.

The following truncated example shows how you would specify an aggregation datalink as the lower link of a zone's `anet` resource during a `zonecfg` interactive session,.

```
# zonecfg -z zone1
.
.
```

```
.
zonecfg:zone1> add anet
zonecfg:zone1:anet> set lower-link=aggr0
.
.
.
zonecfg:zone1:anet> end
zonecfg:zone1> commit
```

For more information about using the `zonecfg` command interactively, see the `zonecfg`(1M) man page and "Creating and Using Oracle Solaris Zones ".

Setting Up an EVS Virtual Tenant Network

A virtual switch is either a software or hardware entity that facilitates inter-virtual machine (VM) communication by looping the inter-VM traffic within a physical machine rather than sending it out on the wire.

EVS enables you to explicitly create virtual switches that span one or more nodes (physical machines), which further virtualizes your network. The virtual switch that you create represents an isolated L2 segment that uses either VLANs or VXLANs to implement the isolation.

For more information about the EVS architecture, see "EVS Components" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

This scenario's overall objective is to set up and deploy an EVS virtual tenant network. The primary objective is to create an elastic virtual switch (`vswitch`) that connects two compute nodes so that both nodes are part of the same L2 segment and can communicate with each other.

The individual objectives for this scenario are as follows:

- Deploy a virtual tenant network with two zones that are connected to the network through `anet` VNICs.
- Deploy the VNICs on a private cloud infrastructure that has two compute nodes.
- Use a VLAN L2 infrastructure to instantiate the private virtual tenant network.

Note - Other L2 technologies such as VXLANs are also supported. For more information, see "Use Case: Configuring an Elastic Virtual Switch for a Tenant" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

The following figures represent the virtual and physical components of the elastic virtual switch configuration that is used in this scenario.

FIGURE 2-1 Virtual Components of an EVS Switch Configuration

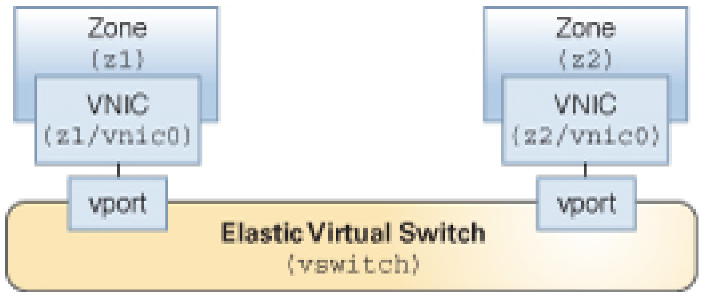
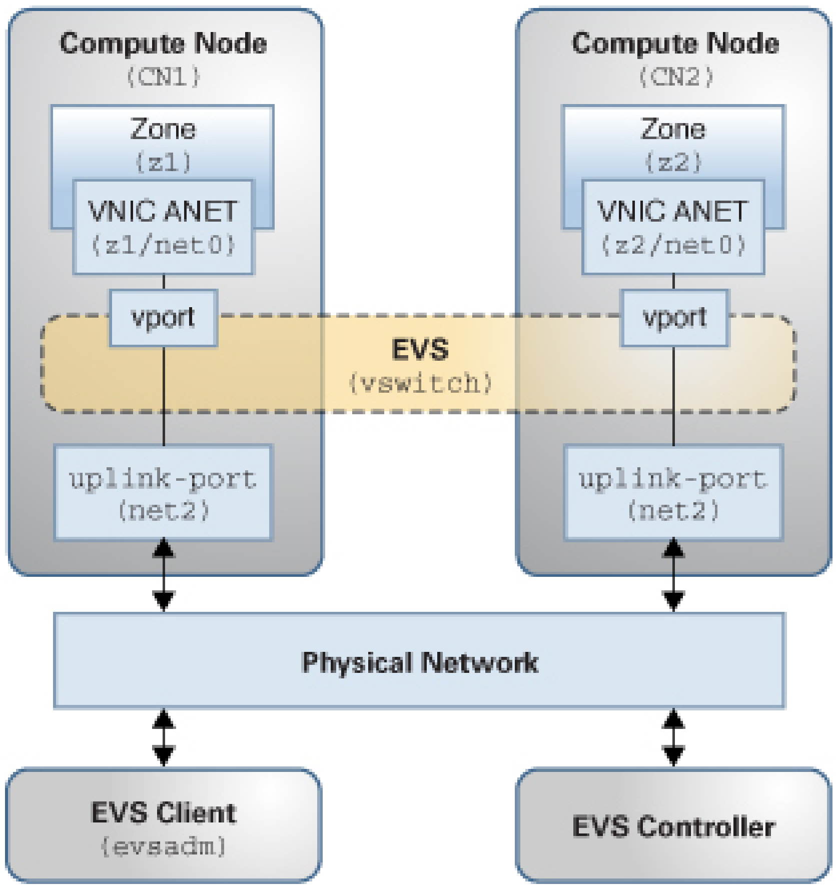


FIGURE 2-2 Physical Components of an EVS Switch Configuration



The following setup uses four network nodes with the following configuration:

- Two compute nodes (CN1 and CN2).
- Two zones (`z1` and `z2`) that are configured on CN1 and CN2, respectively.
- The two zones (`z1` and `z2`) are configured with a VNIC `anet` resource on each zone.
- One node that acts as the EVS controller.
- One node that acts as the EVS client.

Note - The EVS controller and the EVS client can be located on the same host.

- Two uplink-ports (`net2`) that specify the datalink to be used for the VLANs.

Perform Preliminary Tasks Prior to Creating an EVS Virtual Tenant Network

The following one-time setup tasks are described:

- Plan the EVS virtual tenant network deployment by doing the following:
  - Select the two compute nodes.
  - Designate a node to act as the controller.
  - Designate a node to act as the client.

Note - The client and controller nodes can be on the same host.

  - Select the VLAN ID range to be used for tenant traffic.
  - Decide which datalink to use for tenant traffic on each compute node.
- Install the base EVS package (`pkg:/service/network/evs`) on every node.
- Install the `pkg:/system/management/rad/module/rad-evs-controller` package on the controller node.
- Configure each of the nodes so that Remote Administration Daemon (RAD) invocations are enabled.
- On every node, configure EVS to point to the controller.
- From the EVS client node, configure the controller properties.
- From the EVS client node, verify the controller configuration.

EXAMPLE 2-10 Installing Mandatory EVS Packages

Prior to setting up an EVS switch, you need to install the necessary software packages. You install these packages on each EVS node separately.

Install the base EVS package (`pkg:/service/network/evs`) on every node (client, controller, and the compute nodes) as follows:

```
# pkg install evs
```

Install the `pkg:/system/management/rad/module/rad-evs-controller` package on the node that is designated as the EVS controller as follows:

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

After installing the mandatory EVS packages and prior to configuring and setting properties for the EVS controller, you must configure all of the nodes so that RAD invocations between each node can take place. For detailed instructions, see “Security Requirements for Using EVS” in “Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ”.

EXAMPLE 2-11 Configuring and Setting Properties for the EVS Controller

The EVS controller provides the resources that are associated with creating and managing elastic virtual switches. You set properties for the controller that specify the necessary information for implementing L2 segments across physical nodes. See “EVS Controller” in “Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ”.

Configure each compute node so that it points to the EVS controller. This scenario uses two compute nodes, so you will need to run the following command on each of the compute nodes:

```
# evsadm set-prop -p controller=CONTROLLER
```

From the client node, configure the EVS controller properties.

1. Set the L2 topology.

   ```
   # evsadm set-controlprop -p l2-type=vlan
   ```
2. Set the VLAN range.

   ```
   # evsadm set-controlprop -p vlan-range=200-300
   ```
3. Specify the uplink-port (datalinks) that are used for the VLAN.

   ```
   # evsadm set-controlprop -p uplink-port=net2
   ```
4. Verify the controller configuration on the client.

   ```
   # 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        --          --
   ```

Create an EVS Virtual Tenant Network (`vswitch`)

The following examples show how you would set up and configure an EVS virtual tenant network named `vswitch`. Pay particular attention to where you perform each task.

The following configuration tasks are described:

- From the client node, set up a virtual switch.
- On each compute node, create a zone and then connect the zone to the virtual switch.
- From the client node, display the EVS configuration.

For an overview of the EVS feature, see Chapter 5, “About Elastic Virtual Switches,” in “Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ”.

EXAMPLE 2-12 Setting Up an EVS Switch

The follow example shows how you would set up an EVS virtual tenant network. You perform this task from the client node.

First, create the EVS switch, named `vswitch` in this example, as follows:

```
# evsadm create-evs vswitch
```

Add the IPnet information to the EVS switch and verify the configuration.

```
# evsadm add-ipnet -p subnet=192.168.70.0/24 vswitch/ipnet
# evsadm show-ipnet

NAME             TENANT     SUBNET          DEFROUTER    AVAILRANGE
vswitch/ipnet    sys-global 192.168.70.0/24 192.168.70.1 192.168.70.2-192.168.70.254
```

Verify that the EVS switch was successfully created.

```
# evsadm
NAME         TENANT       STATUS     VNIC      IP              HOST
vswitch      sys-global    --         --       vswitch_ipnet   --
```

Check the VLAN ID that is associated with the virtual switch.

```
# evsadm show-evs -L
EVS        TENANT      VID      VNI
vswitch    sys-global  200      --
```

EXAMPLE 2-13 Creating and Connecting a Zone to an EVS Switch

The following example describes how you would create a zone on each tenant, and then connect the zone to the virtual switch.

On each tenant, configure a zone with an `anet` VNIC resource as follows:

```
# zonecfg -z z1
zonecfg:z1> create
.
.
.
zonecfg:z1> add anet
zonecfg:z1:anet> set evs=vswitch
zonecfg:z1:anet> end
zonecfg:z1> commit
zonecfg:z1> exit
```

For more information, see "Creating a VNIC anet Resource for an Elastic Virtual Switch" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

For information about setting `anet` resource properties that pertain to an EVS switch, see "Resource Types and Properties" in "Introduction to Oracle Solaris Zones ".

Boot the zone.

```
# zoneadm -z z1 boot
```

Verify that the VNIC was created and is connected to the virtual switch.

```
# dladm show-vnic -c
LINK        TENANT        EVS       VPORT        OVER    MACADDRESS        VIDS
z1/net0     sys-global    vswitch   sys-vport0   net2    2:8:20:1a:c1:e4   200
```

From within the zone, verify that the IP address has been assigned.

```
# 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         --       192.168.84.3/24
```

From the client node, display the EVS configuration.

```
# evsadm
NAME      TENANT      STATUS      VNIC      IP      HOST
vswitch    sys-global -- --       vswitch_ipnet
```

EVS provides a rich set of functionality that is not fully described in this scenario. For additional tasks and use cases, see Chapter 6, “Administering Elastic Virtual Switches,” in “Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ”.

Combining Network Virtualization With Oracle VM Server for SPARC to Create a Cloud Environment

The following scenario combines network virtualization features with Oracle VM Server for SPARC to create a multilevel virtual network that parallels a cloud environment. This deployment method provides highly efficient, enterprise-class virtualization capabilities for Oracle's SPARC T-Series servers and supported M-Series servers.

This scenario assumes that you are running an Oracle VM Server for SPARC version that supports Oracle Solaris 11.2. For more information about Oracle VM Server for SPARC, refer to the documentation library at http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html.

At a high level, the objective of this scenario is to carve up a SPARC based system into multiple Oracle Solaris VM Server guest domains, where each domain corresponds to a node within a cloud environment. You can deploy per-tenant workloads as zones within these Oracle VM Server for SPARC guest domains.

Configuring network virtualization features in this way enables you to build an entire cloud within a single SPARC based system. Or, you can use this type of configuration to integrate a SPARC based system into a larger cloud environment, where the system appears as a set of nodes within that environment.

Combining network virtualization features with Oracle VM Server for SPARC parallels a traditional cloud in the following ways:

- Compute nodes are implemented as Oracle VM Server for SPARC guest domains.
- Compute nodes communicate with each other through the virtual network infrastructure that is provided by Oracle VM Server for SPARC and Oracle Solaris 11 running on the service domain.
- The `vnet` driver instances that are within each guest domain correspond to a physical NIC within a physical compute node.

The benefits of this type of configuration include the following:

- Enables more flexibility by enabling you to run smaller domains that you can upgrade individually without affecting other workloads that are running on the system.
- Takes advantage of SPARC Reliability, Availability, and Serviceability (RAS) features.
- Uses a faster virtual network for communication between nodes instead of relying on a physical infrastructure.

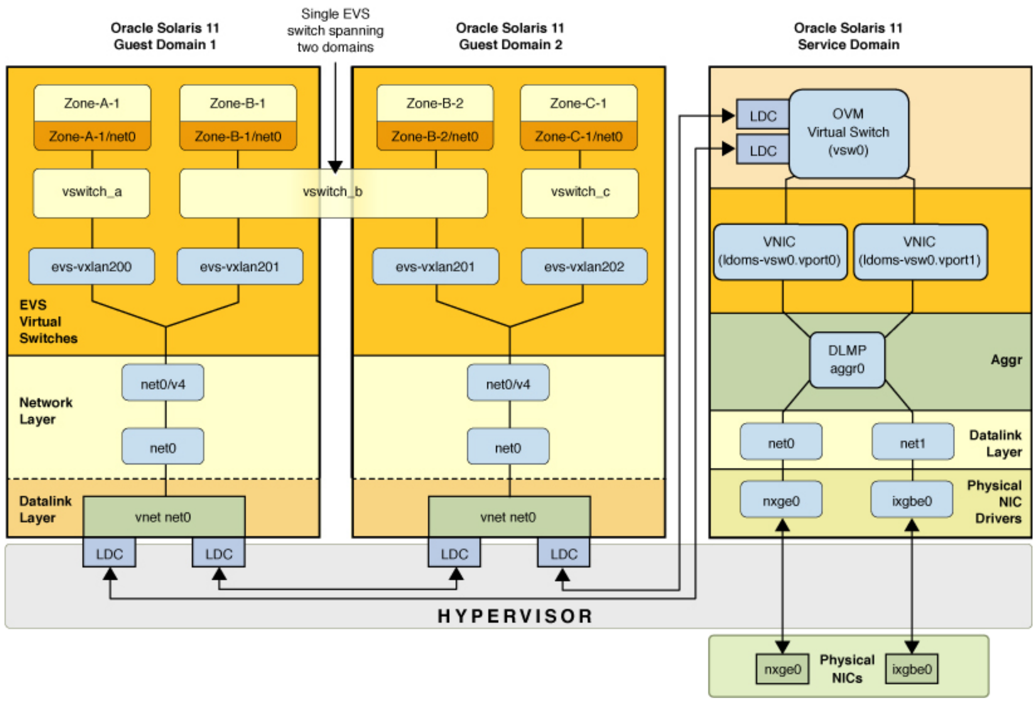
Objectives for Creating and Deploying a Cloud Environment

The deployment objectives for this scenario are as follows:

- Configure a virtual network on an Oracle VM Server for SPARC service domain.
- Configure two Oracle VM Server for SPARC guest domains to be used as containers for multiple zones that are configured within each guest.
- Have each guest domain then correspond to a specific compute node within the cloud that will run the various workloads.
- Configure the elastic virtual switches that will be used to connect the zones running in the guest domains.
- Carve up the guest domains into multiple zones that will run the various workloads.

The following figure illustrates the two distinct levels of network virtualization that you create with this configuration.

FIGURE 2-3 Combining Network Virtualization Features With Oracle VM Server for SPARC



On the first level, you configure network virtualization features that are supported by Oracle VM Server for SPARC. This part of the network virtualization combines Oracle VM Server for SPARC configuration with the Oracle Solaris 11 OS that is running on the service domain. The `vnet` configuration takes place at this first level of virtualization. Because the configuration only relies upon IP connectivity from the guest domains, no additional support from Oracle VM Server for SPARC is required for the configuration on the second network virtualization level to work.

On the second level, EVS is used to create elastic virtual switches across the guest domains. EVS is configured to use the `vnet` interfaces as uplinks. VXLAN datalinks are automatically created by EVS from each guest domain and then used to encapsulate the traffic of the individual elastic virtual switches.

The figure represents the following configuration:

- Two physical NICs, `nxge0` and `ixgbe0`, which are directly assigned to the service domain where they are represented by datalinks `net0` and `net1`.
- To provide high availability in case of failure of the physical NICs, `net0` and `net1` in the service domain are grouped into the DLMP aggregation (`aggr0`).

- The aggregation, `aggr0`, is then connected to an Oracle VM Server for SPARC virtual switch in the service domain named `vsw0`.

  Two VNICs, `ldoms-vsw.vport0` and `ldoms-vsw.vport1`, are automatically created by `vsw0`, with each VNIC then corresponding to the Oracle VM for SPARC `vnet` instances within the guest domains.
- The `vsw0` and the `vnet` instances communicate with each other through the hypervisor by using Logical Domain Channels (LDCs).
- Each guest uses its instance of the `vnet0` driver, which appears in the guest domain as a datalink (`net0`) for the purpose of communicating with other guest domains and the physical network.
- In each guest domain, the `vnet` datalinks (`net0`) are configured with the IP interface `net0/v4`.
- Each guest domain is an EVS compute node, with three EVS switches, `vswitch_a`, `vswitch_b`, and `vswitch_c`, that are configured from the EVS controller (not shown in this figure).
- EVS is configured to use a VXLAN as its underlying protocol. For each guest domain that uses an elastic virtual switch, EVS automatically configures a VXLAN datalink. These VXLAN datalinks are named `evs-vxlan`*id*, where *id* is the VXLAN ID that is assigned to the virtual switch.
- In the guest domains, Oracle Solaris zones are configured to run the tenants' workload. Each zone is connected through a VNIC and a virtual port (not shown in this figure) to one of the EVS switches.
- `Zone-B1` and `Zone-B2` belong to the same user and are running on two different guest domains. The EVS switch, `vswitch_b`, is instantiated on both guest domains. To the two zones, it appears as if each zone is connected to a single Ethernet segment that is represented by `vswitch_b` and isolated from the other virtual switches.
- EVS automatically creates the VXLAN datalinks that are needed by the various elastic virtual switches. For example, for `vswitch_b`, EVS automatically created a VXLAN datalink named `evs-vxlan201` on each of the guest domains.

Configure a Virtual Network on the Oracle VM Server for SPARC Service and Guest Domains

Perform the following configuration tasks:

- On the service domain, create and configure a DLMP aggregation.
- On the service domain, configure the Oracle VM Server for SPARC virtual switch.
- On the service domain, configure the Oracle VM Server for SPARC virtual network devices for use on the guest domains
- On each of the guest domains, configure an IP address for each `vnet`.

The following examples assume that you have already configured your Oracle VM Server for SPARC (formerly called Sun Logical Domains, or LDoms) infrastructure with a control domain and a service domain and that you have created two guest domains to be used as cloud nodes.

For step-by-step instructions on setting up your Oracle VM Server for SPARC infrastructure, see the white papers at http://www.oracle.com/technetwork/server-storage/vm/overview/index.html.

Note - The examples that describe this scenario are presented in the order in which you would perform each of the individual tasks.

EXAMPLE 2-14 Creating and Configuring a DLMP Aggregation

The following example describes the first configuration task in this scenario, which is to create a DLMP aggregation on the Oracle VM Server for SPARC service domain. In this example, you create the DLMP aggregation (`aggr0`) with probing enabled for the `net1` and `net2` interfaces.

```
servicedomain# dladm create-aggr -l net1 -l net2 -m dlmp -p probe-ip=+ aggr0
```

For additional information, see Example 2-8.

EXAMPLE 2-15 Creating an Oracle VM Server for SPARC Virtual Switch

One of the basic components that Oracle VM Server for SPARC uses for virtual networking is a virtual switch (`vsw`). A virtual switch is similar to an Ethernet switch, in that it runs in an I/O or service domain and switches Ethernet packets over Logical Domain Channels (LDCs), as well using the Oracle Solaris 11 built-in virtual switch.

The following example shows how to create a virtual switch on the DLMP link portion of the configuration. You perform this task on the service domain.

```
servicedomain# ldm add-vsw net-dev=aggr0 primary-vsw0 primary
```

For more information about configuring virtual switches for Oracle VM Server for SPARC, see “Virtual Switch” in “Oracle VM Server for SPARC 3.1 Administration Guide ”.

EXAMPLE 2-16 Creating Virtual Network Devices for Oracle VM Server for SPARC Guest Domains

The second basic component that Oracle VM Server for SPARC uses for virtual networking is a virtual network device (`vnet`). A virtual network device is plumbed into the guest domain.

The following example shows the next configuration task in this scenario, which is to create a virtual network device for each guest domains. You also perform this task on the service domain.

```
servicedomain# ldm add-vnet
```

You create one virtual network device, per guest domain. For every device that you create, a `vnet` instance is also created in the corresponding guest domain.

Next, configure an IP address for each `vnet` on each guest domain as follows:

```
guestdomain1# ipadm create-ip net0
guestdomain# ipadm create-addr -t -a 192.168.70.1 net0


guestdomain2# ipadm create-ip net0
guestdomain# ipadm create-addr -t -a 192.168.70.2 net0
```

For more information about creating virtual network devices, see "Virtual Network Device" in "Oracle VM Server for SPARC 3.1 Administration Guide ".

Create an EVS Switch to Deploy the Cloud Workload

The next group of tasks involves creating an EVS switch that you use to deploy the cloud workload. Some configuration tasks are performed on the Oracle VM Server for SPARC service domain, while others are performed on the guest domains.

The following EVS setup is used:

- Two compute nodes that correspond to two guest domains. Each guest domain has a `net0` interface for its `vnet` datalink, which are then used as `uplink-ports` by the elastic virtual switch.
- One node that acts as the EVS controller.
- One node that acts as the EVS client.

Note - The EVS controller and the EVS client can be located on the same host.

- Set of four zones: `Zone-A1` and `Zone-B1`, which are configured on the first guest domain, and `Zone-B2` and `Zone-C2`, which are configured on second guest domain.
- The four zones are configured with a VNIC (`anet`) resource on each zone, which are then connected to the EVS switches.

▼ How to Configure an EVS Virtual Switch to Deploy a Cloud Workload

Before You Begin Perform all of the necessary planning and prerequisite tasks, which include installing the EVS packages and configuring the appropriate authorizations.

For planning instructions, see "Perform Preliminary Tasks Prior to Creating an EVS Virtual Tenant Network" on page 36.

For security requirements, see "Security Requirements for Using EVS" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

1. **Configure each compute node to point to the EVS controller.**

   ```
   # evsadm set-prop -p controller=CONTROLLER
   ```

   You can deploy the EVS controller on any node, as long as it can reach the Oracle VM Server for SPARC guest domains through the `vnet` interfaces.

   For example, you can deploy the EVS controller in any of the follow ways:

   - In the service domain's global zone
   - In a non-global zone of the service domain
   - In its own guest domain
   - On a separate physical machine

   For more information about configuring an EVS controller, see "Configuring an EVS Controller" in "Managing Network Virtualization and Network Resources in Oracle Solaris 11.2 ".

2. **On the controller, configure the required EVS properties for the compute nodes.**

   a. **Set the L2 topology.**

      ```
      # evsadm set-controlprop -p l2-type=vxlan
      ```

   b. **Set the VXLAN range and IP address.**

      ```
      # evsadm set-controlprop -p vxlan-range=200-300
      # evsadm set-controlprop -p vxlan-addr=192.168.70.0/24
      ```

      Note that you determine the VXLAN range during the planning phase, prior to setting up the EVS switch. For more information about configuring EVS controller properties, see Example 2-11.

   c. **Specify the uplink-port (datalinks) that are used for the VXLAN.**

      ```
      # evsadm set-controlprop -p uplink-port=net0
      ```

   d. **Verify the configuration.**

      ```
      # evsadm show-controlprop -p l2-type,vxlan-range,vxlan-addr
      NAME                VALUE               DEFAULT             HOST
      l2-type             vxlan               vxlan               --
      ```

```
vxlan-range         200-300           --                  --
vxlan-addr          192.168.70.0/24   0.0.0.0             --
uplink-port         net0              --                  --
```

The controller must be assigned an IP address that is reachable from each of the guest domains. For this example, that IP address is 192.168.70.10.

3. **Create and verify the EVS virtual switch, which is named `vswitch_a` in this example.**

   a. **Create the EVS switch.**

   ```
   # evsadm create-evs vswitch_a
   ```

   Repeat this step to create the two other EVS switches that are used in the configuration (`vswitch_b` and `vswitch_c`).

   b. **Add the IPnet information to the EVS switch and verify the configuration.**

   ```
   # evsadm add-ipnet -p subnet=192.168.80.0/24 vswitch_a/ipnet
   # evsadm show-ipnet

   NAME             TENANT     SUBNET          DEFROUTER    AVAILRANGE
   vswitch_a/ipnet  sys-global 192.168.80.0/24 192.168.80.1 192.168.80.2-192.168.80.254
   ```

   Repeat this step for the two other EVS switches that are used in the configuration (`vswitch_b` and `vswitch_c`).

   c. **Verify that the virtual switch was successfully created.**

   ```
   # evsadm
   NAME         TENANT       STATUS     VNIC      IP               HOST
   vswitch_a    sys-global    --         --      vswitch_a/ipnet    --
   ```

   d. **Check the VLAN ID that is associated with the virtual switch.**

   ```
   # evsadm show-evs -L
   EVS       TENANT        VID      VNI
   vswitch_a   sys-global  --       200
   vswitch_b   sys-global  --       201
   vswitch_c   sys-global  --       202
   ```

Create Oracle Solaris Zones on the Oracle VM Server for SPARC Guest Domains

The following example shows how to create a zone within an Oracle VM Server for SPARC guest domain for the purpose of deploying the cloud workload. The following commands create a zone on the guest domain with an `anet` that uses a VXLAN as the underlying link of the Oracle VM Server for SPARC virtual switch.

```
# zonecfg -z B-1
zonecfg:B-1> create
.
.
.
zonecfg:B-1> add anet
zonecfg:B-1:anet> set evs=vswitch_b
zonecfg:B-1:anet> end
zonecfg:B-1> commit
zonecfg:B-1> exit
```

For more information about configuring zones, see “Creating and Using Oracle Solaris Zones ”.

◆ ◆ ◆ CHAPTER 3

Oracle Solaris Network Administration Command Cheatsheet

This chapter provides a quick reference to basic commands that are used for network administration when in the fixed mode. Fixed mode is primarily used to administer network configuration in an enterprise environment.

For information about the commands that you use to administer network configuration in the reactive mode (most often used for notebook PCs), see Chapter 6, "Administering Profile-Based Network Configuration in Oracle Solaris," in "Configuring and Administering Network Components in Oracle Solaris 11.2 ".

Network Administration Command Cheatsheet

The following command reference describes how to perform common network administration tasks in the Oracle Solaris release. For more information about these commands, see the `dladm`(1M), `ipadm`(1M), and `route`(1M) man pages.

Note - The various parameters that are specified in the following tasks are provided as examples *only*. The parameters that you specify will most likely vary from those that are used in this quick reference.

List all of the profiles on a system:

```
netadm list
```

Switch to the fixed mode by enabling the `DefaultFixed` profile:

```
# netadm enable -p ncp DefaultFixed
```

Display all of the datalinks (physical and virtual) on a system:

```
# dladm show-link
```

Display all of the physical datalinks on a system:

```
# dladm show-phys
```

Display all of the properties for all of the datalinks on a system:

```
# dladm show-linkprop
```

Display all of the properties for a specific datalink on a system:

```
# dladm show-linkprop net0
```

Display a specific property for a specific datalink on a system:

```
# dladm show-linkprop -p mtu net0
```

Modify a property of a specific datalink on a system, for example the MTU value:

```
# dladm set-linkprop -p mtu=1500 net0
```

Reset the default value for the property of a specific datalink on a system:

```
# dladm reset-linkprop -p mtu net0
```

Display general information about a system's interfaces:

```
# ipadm
```

The output of this command is comparable using the `ifconfig` command to obtain similar information.

Display a system's IP interfaces and addresses, including the netmask if set:

```
# ipadm show-addr
```

Create an IP interface and then configure a static IPv4 address for that interface:

```
# ipadm create-ip net0
# ipadm create-addr -a local=10.9.8.7/24 net0/addr
```

Obtain an IP address from a DHCP server:

```
# ipadm create-ip net0
# ipadm create-addr -T dhcp net0/addr
```

Create an auto-generated IPv6 address:

```
# ipadm create-ip net0
# ipadm create-addr -T addrconf net0/addr
```

Change the netmask property for an IP address object name (net3/v4) to 8:

```
# ipadm set-addrprop -p prefixlen=8 net3/v4
```

Configure a persistent default route for a system:

```
# route -p add default 192.168.1.1
```

Configure a static route for a system:

```
# route -p add -net 192.168.3.0 -gateway 192.168.1.1
```

Configure a system's host name (`myhost`):

```
# hostname myhost
```

Configure DNS on a system:

```
# svccfg -s dns/client setprop config/nameserver=net_address: 192.168.1.1
# svccfg -s dns/client setprop config/domain = astring: "myhost.org"
# svccfg -s name-service/switch setprop config/host = astring: "files dns"
# svcadm refresh name-service/switch
# svcadm refresh dns/client
```

Index

D

E

O

P

R

W

Z