5

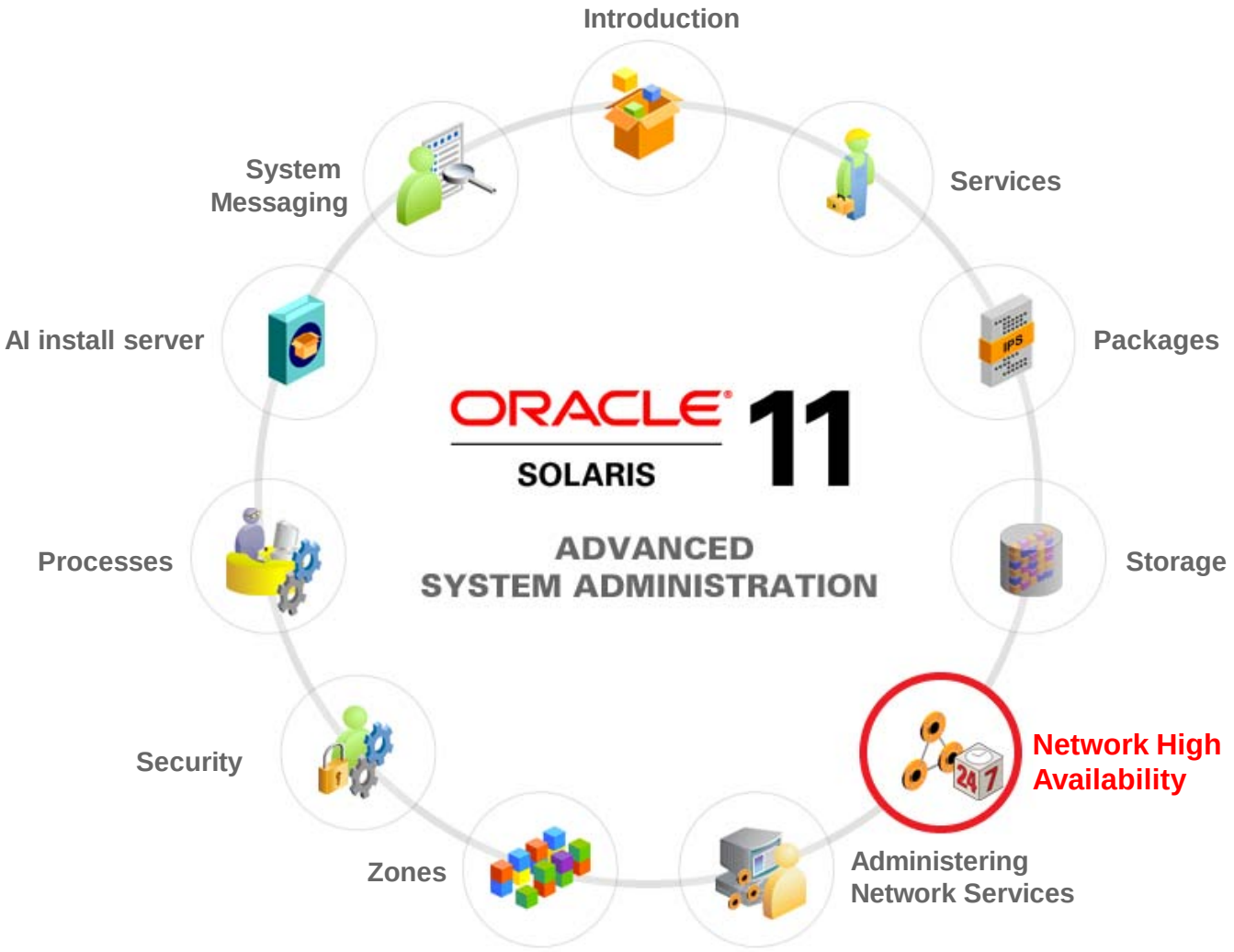
# Configuring the Network

# Objectives

After completing this lesson, you should be able to configure:

- A virtual switch
- Link aggregation for high performance
- IPMP for IP high availability
- Packet Filter to control network access

# Job Workflow



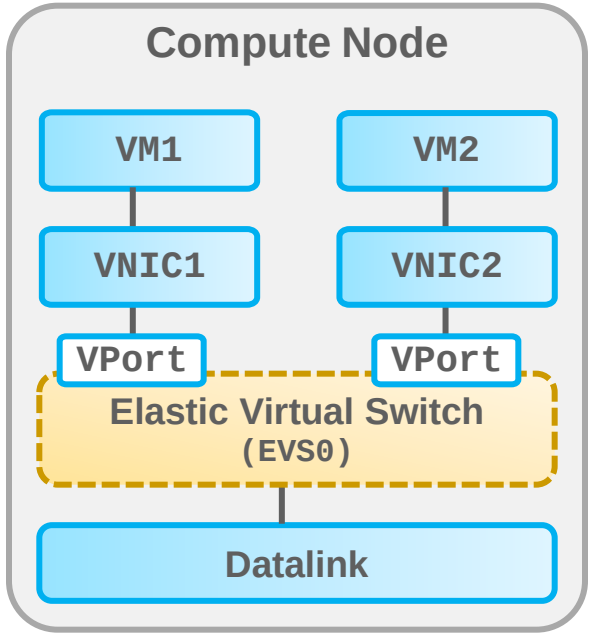
# Agenda

- Configuring virtual switches
- Configuring link aggregation for high performance
- Configuring IPMP for IP high availability
- Configuring Packet Filter to control network access

# Elastic Virtual Switch: Overview

An elastic virtual switch (EVS):

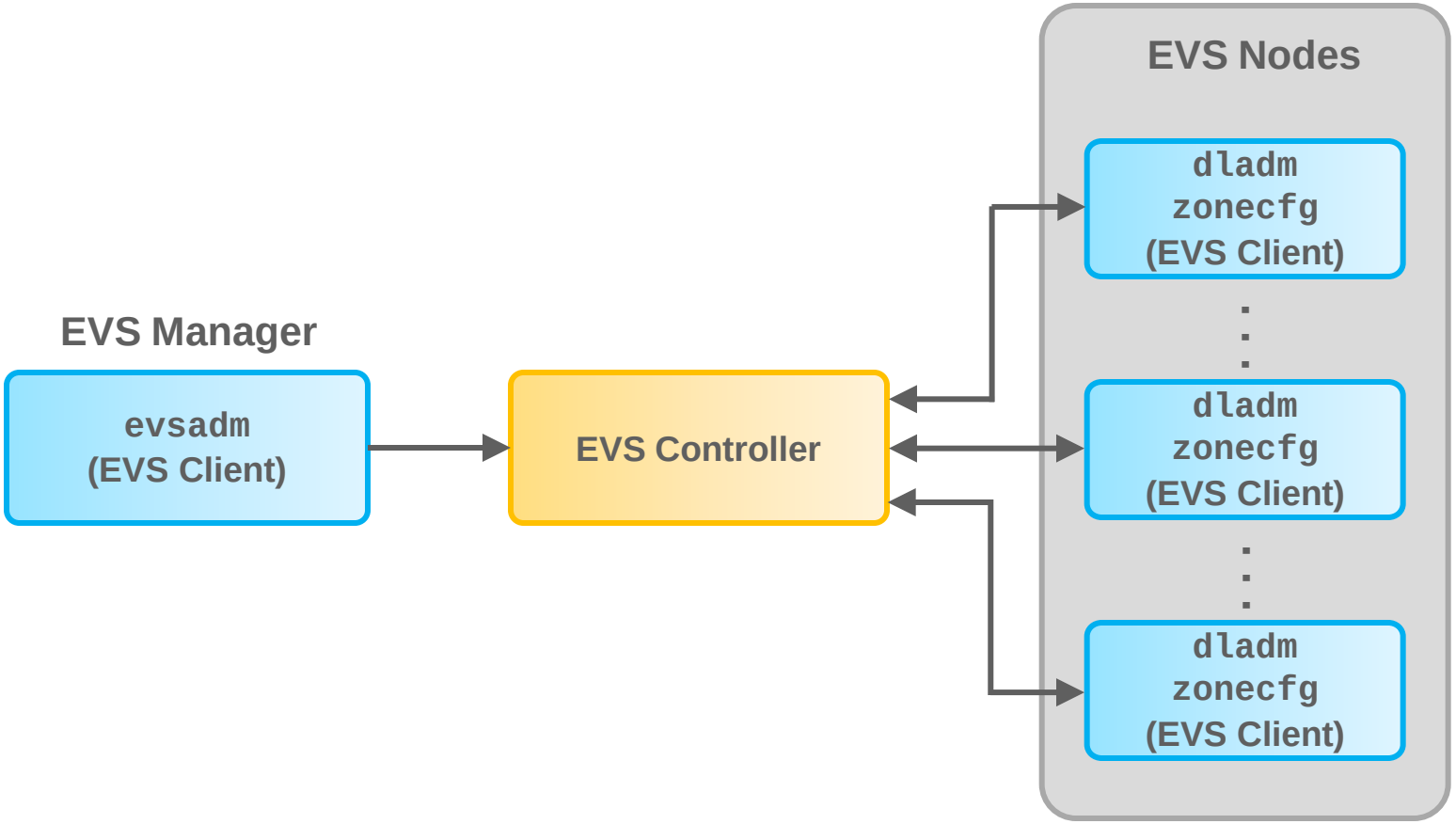
- Is an entity that represents explicitly created virtual switches that belong to the same Layer 2 (L2) segment
- Enables you to create and administer a virtual switch that spans one or more physical machines (nodes)
- Provides network connectivity between VMs connected to it from anywhere in the network



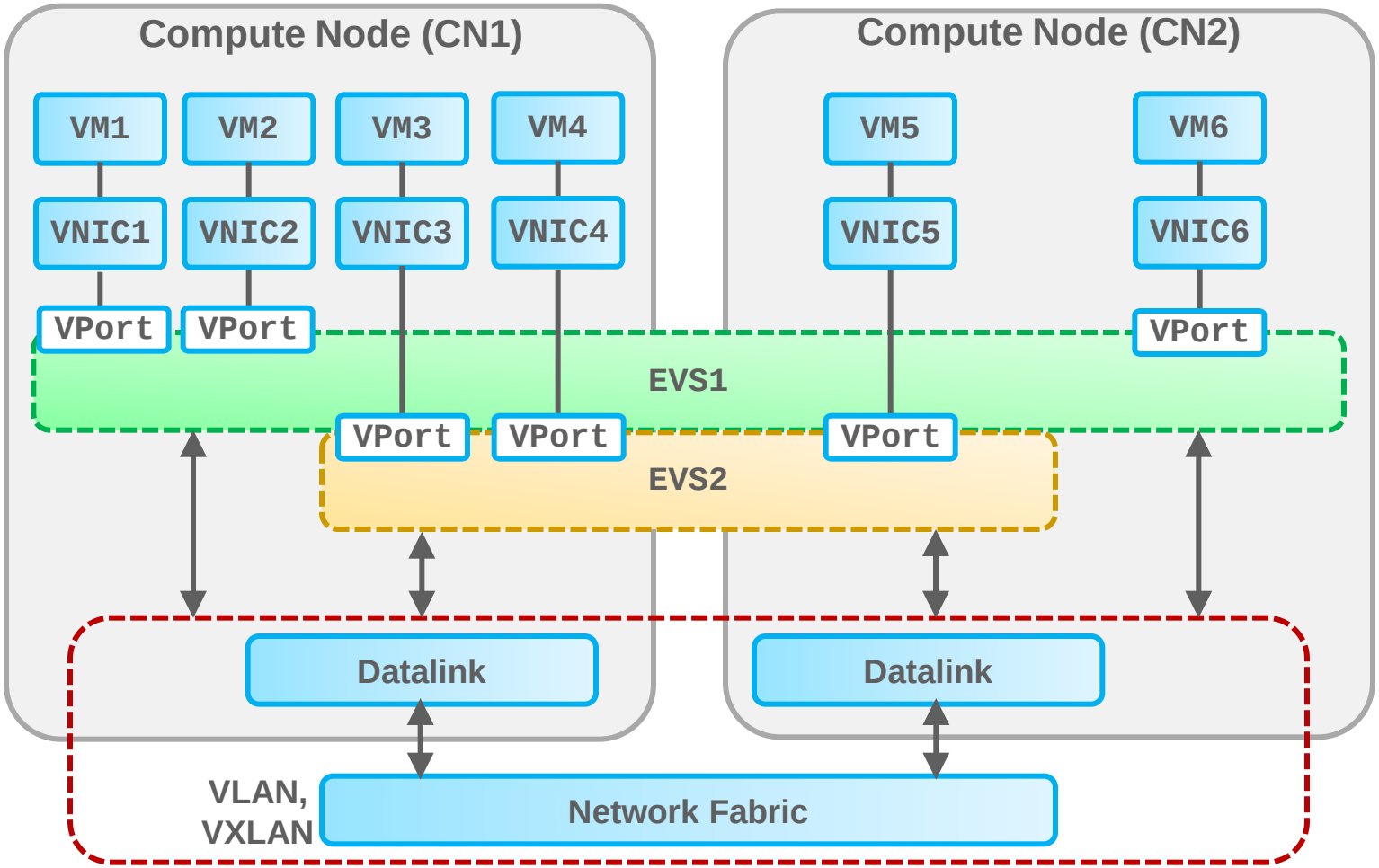
# Benefits of Using the EVS Feature

- Creates a virtual network between VMs that are on multiple servers, thus providing network connectivity
- Supports addition of virtual ports with custom SLAs or profiles
- Provides network isolation by using VLANs or VXLANs
- Supports multitenant virtual networks that share the same underlying infrastructure
- Provides centralized management of:
  - MAC address and IP address for the virtual ports
  - SLAs on a per-virtual-switch or per-virtual-port basis
  - Monitoring the runtime network traffic statistics of the virtual ports
- Is integrated with Oracle Solaris Zones and Oracle Solaris Kernel Zones

# EVS Components



# EVS: Example



# EVS Administrative Commands

| Commands | Description |
|---|---|
| **evsadm** | You use the `evsadm` command to communicate with the EVS controller and manage the elastic virtual switch, IPnet, and VPorts. |
| **evsstat** | You use the `evsstat` command to display the network traffic statistics for all the VPorts in a data center or for all the VPorts of the specified elastic virtual switch. |
| **dladm** | You can administer the VNICs connected to an elastic virtual switch by using the following `dladm` commands:<br>• **`create-vnic`:** Enables you to create a VNIC and specify the elastic virtual switch name to which you must connect the VNIC. Optionally, you can specify the VPort of the elastic virtual switch.<br>• **`show-vnic`:** Enables you to display the EVS information for a specific VNIC. The output of the `dladm show-vnic` command also displays the fields `TENANT`, `EVS`, and `VPORT`. However, these fields are not visible from within a zone. |
| **zonecfg** | You use the enhanced `zonecfg` command to configure a zone's VNIC `anet` resource for an elastic virtual switch. |

# Planning an EVS Configuration

1. Install the mandatory EVS packages on the EVS controller, EVS manager, and EVS nodes.
2. Set up EVS authentication with the preshared public key for `evsuser`:
   - From the EVS manager to the EVS controller
   - From the EVS controller to each EVS node
   - From each EVS node to the EVS controller
3. Specify the EVS controller by setting the `controller` property. You must specify the host name or IP address of the EVS controller on the EVS nodes, EVS manager, and EVS controller.

# Planning an EVS Configuration

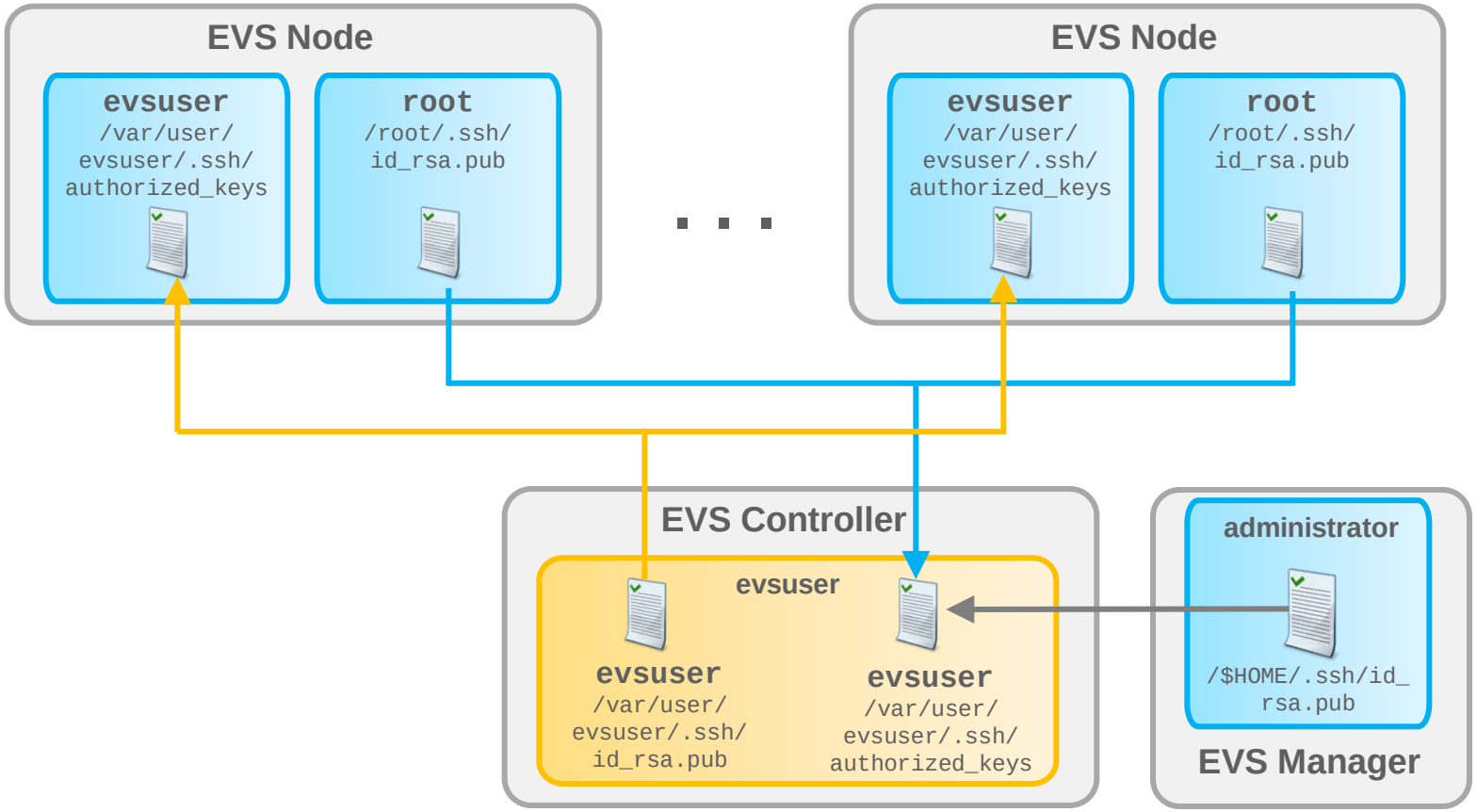
4. Configure the EVS controller, during which you must consider the following:
   - Determine whether you are implementing the elastic virtual switch by using a VLAN, VXLAN, or both.
     - If you use a VLAN, you must set the properties `uplink-port` and `vlan-range`.
     - If you use a VXLAN, you must set the properties `vxlan-range` and `uplink-port` or `vxlan-addr`.
   - If the compute nodes do not have the same datalink, then for every compute node, you must specify the datalink for the `uplink-port` property.
5. Configure the elastic virtual switch by using the EVS manager.
6. Create VNICs on the EVS nodes and connect the VNICs to the elastic virtual switch.

# Installing the Mandatory EVS Packages

You must install the following packages before using EVS:

| Packages | Description |
|---|---|
| `pkg:/service/network/evs` | You must install this core package on the EVS manager, EVS controller, and EVS nodes. This package contains the following components:<br>• `evsadm`<br>• `evsstat`<br>• `svc:/network/evs:default` |
| `pkg:/system/management/rad/module/rad-evs-controller` | You must install this package only on the system that acts as an EVS controller. This package contains the SMF service named `svc:/network/evs-controller:default`. |

# Setting Up SSH Authentication




**Note:** The assumption here is that the `controller` property is set to `ssh://evsuser@evs-controller.example.com` on each host.

# Configuring an EVS Controller

1. Set the EVS controller.

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

2. Display the configured EVS controller.

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

3. Set the properties for the EVS controller.

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

4. Display the properties of the EVS controller.

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

# Configuring an EVS

1. Create an elastic virtual switch.

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

2. Add an IPnet to the elastic virtual switch.

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

3. Add a VPort to the elastic virtual switch.

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

4. Display the configured elastic virtual switch.

```
# evsadm
```

# Creating VNICs for an Elastic Virtual Switch

1. Configure a VNIC for an elastic virtual switch.

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

2. Display information about the VNICs that are connected to an elastic virtual switch.

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

# EVS and Zones

- Oracle Solaris Zones and Oracle Solaris Kernel zones support the EVS feature.
- Kernel zones support VNICs that you create for EVS.
- The VNIC that is created in the Kernel zone works only if the VNIC uses the factory MAC addresses that are associated with the `zvnet` driver.
- In the Kernel zone, you can connect the VNIC to the VPort that is created by using the `evsadm add-vport` command.

# EVS and Zones

- For an `anet` resource that connects to an EVS with the `evs` and `vport` properties set, the properties of that `anet` resource are encapsulated in the `evs` and `vport` pair.
- You can also set the `tenant` resource if you have configured a tenant for an EVS.
- You can set the following properties for an EVS `anet` resource:
  - `linkname`
  - `evs`
  - `vport`
  - `configure-allowed-address`

# Creating a VNIC anet Resource or an EVS

This example shows you how to create a zone that has a VNIC anet resource evszone/net1, which is connected to ORA and vport0 of the tenant tenantA.

```
# 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
# dladm show-vnic -c
LINK         TENANT    EVS VPORT    OVER   MACADDRESS        VIDS
evszone/net1 tenantA   ORA vport0   net2   2:8:20:89:a1:97   200
```

# Quiz



You do not need to set up SSH authentication between the EVS manager and EVS nodes.

a. True
b. False

# Quiz



Which of the following commands should you use to administer EVS clients? (Choose two.)

a. zonecfg

b. evsadm

c. dladm

d. evsstat

# Practice 5-1 Overview: (Demonstration) Configuring EVS

In this practice, you watch an EVS configuration demonstration and observe the following processes:

- Setting up SSH authentication
- Configuring the EVS controller, EVS manager, and EVS nodes
- Configuring nonglobal zones to use elastic virtual switches

**Note:** This practice comprises only the demonstration.

# Agenda

- Configuring virtual switches
- Configuring link aggregation for high performance
- Configuring IPMP for IP high availability
- Configuring Packet Filter to control network access

# Importance of Network High Availability

Network high availability is required to ensure that:

- Network needs of the business and the user community are supported
- Network communications remain uninterrupted
- Network performance is good

# What Does 99.99% Mean?

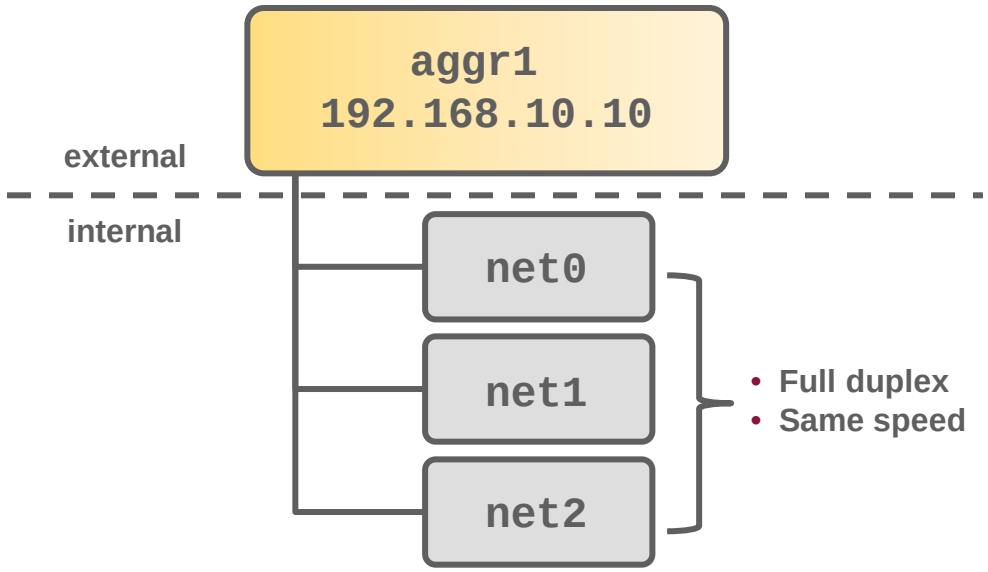
- Availability is usually expressed as a percentage of uptime in a given year, and is referred to as 5 nines, 6 nines, and so on.
- Uptime is a function of down time and recovery time.
- To achieve high availability, you must have low down time and low recovery time.
- So what does 99.99% mean?
  - 99.99% (referred to as 4 nines), means the following:
    - `4 nines -> (365×24) - .9999(365×24) = 8760 - 8759.124 = 0.876 hours = 52 min and 30 secs`
  - 99.99% implies a down time of 52.30 minutes per year.

# High Availability: Overview

- The term *high availability* (HA) refers to a state in which a point of failure is instantly taken over by a redundant system to ensure business continuity.
- HA is architected into domains such as server, storage, applications, and network.
  - In the network domain, HA can be implemented at various levels such as link, IP, and router.
- The following are some of the HA features of Oracle Solaris 11:
  - Link aggregation
  - IP Network Multipathing (IPMP)

# Link Aggregation: Overview

- Link aggregation enables multiple network interface cards (NICs) to be grouped into a single logical interface.
- Link aggregations are useful for increasing bandwidth as well as providing HA.
  - Links must be of the same speed, full duplex, and point-to-point.
  - You use the `dladm` command.



# Link Aggregation Types

Based on single or multiple switch capability, link aggregation can be of two types:

- **Trunk aggregation:** Works only with a single switch
- **Datalink Multipathing (DLMP) aggregation:** Spans multiple switches
  - For DLMP aggregation, no switch-side configurations are required.
  - Switches are therefore unaware of the link aggregation and treat each port individually.

# Aggregation Modes and Switches

LACP switch modes:

- **Off:** Default mode; no LACPDUs
- **Active:** LACPDUs at specified regular intervals
- **Passive:** LACPDUs only when received from switch

# Load Balancing and Aggregation Policies

In policy making, determination of the outgoing link is done by hashing the specific header of each packet:

- **L2 (Networking):** MAC header
- **L3 (Addressing):** IP header
- **L4 (Communication):** TCP/UDP or other ULP header

# Commands to Administer Link Aggregation

| Command | Description |
|---|---|
| `dladm create-aggr` | Create a link aggregation. |
| `dladm add-aggr` | Add a link to an aggregation. |
| `dladm show-aggr -x` | Display link aggregation details. |
| `dladm delete-aggr` | Delete a link aggregation. |
| `dladm remove-aggr` | Remove a link from an aggregation. |
| `dladm modify-aggr` | Switch between link aggregation types or modify a trunk aggregation. |

# Preparing for Link Aggregation

Before configuring the link aggregation:

1. Make sure that the links to be combined are full-duplex and point-to-point, and that they operate at identical speeds.
2. Use the `dladm show-link` command to verify state.

   **Note:** If an IP interface is created over the datalink, remove the IP interface first.

```
# dladm show-link
LINK            CLASS   MTU     STATE           OVER
net0            phys    1500    unknown         --
net1            phys    1500    unknown         --
net2            phys    1500    unknown         --
net3            phys    1500    unknown         --
```

# Creating Link Aggregation

Use the following commands to create and display link aggregation:

- `dladm create-aggr`
- `dladm show-aggr`

```
# dladm create-aggr -l net0 -l net1 aggr1
# dladm show-link
LINK      CLASS     MTU  STATE   OVER
net0      phys      1500 up      --
net1      phys      1500 up      --
net2      phys      1500 unknown --
net3      phys      1500 unknown --
aggr1     aggr      1500 up      --     net0 net1
# dladm show-aggr
LINK     MODE  POLICY  ADDRPOLICY  LACPACTIVITY     LACPTIMER
aggr1    trunk L4      auto        off              short
```

# Modifying Link Aggregation

Use the following commands to modify link aggregation:

- `dladm modify-aggr`
- `dladm add-aggr`
- `dladm remove-aggr`

```
# dladm modify-aggr --policy=L3 aggr1
# dladm add-aggr -l net2 -l net3 aggr1
# dladm remove-aggr -l net0 aggr1
```

# Deleting Link Aggregation

Use the following command to delete aggregation:

`dladm delete-aggr`

```
# dladm delete-aggr aggr1
```

# Creating a DLMP Aggregation

```
# dladm create-aggr -m dlmp -l net0 -l net1 -l net2 -l net3 speedway0
# dladm show-link
LINK      CLASS     MTU      STATE   OVER
net0      phys      1500     up      --
net1      phys      1500     up      --
net2      phys      1500     up      --
net3      phys      1500     up      --
speedway0 aggr      1500     up      net0 net1 net2 net3
# dladm show-aggr
LINK       MODE  POLICY   ADDRPOLICY  LACPACTIVITY  LACPTIMER
speedway0  dlmp  --       --          --            --
```

# Quiz

Which of the following commands should you use to display the link aggregation details?

a. `dladm show-link –x`

b. `dladm show-aggr –x`

c. `dladm list-aggr -x`

# Practice 5-2 Overview: Configuring a Link Aggregation

This practice covers the following topics:

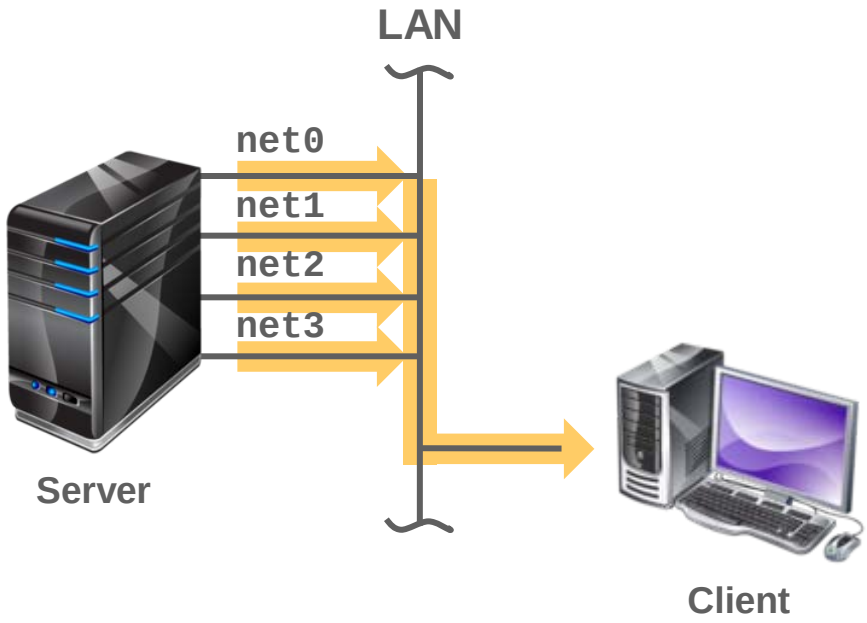
- Creating a link aggregation
- Removing a link aggregation

# Agenda

- Configuring virtual switches
- Configuring link aggregation for high performance
- Configuring IPMP for IP high availability
- Configuring Packet Filter to control network access

# IPMP: Introduction

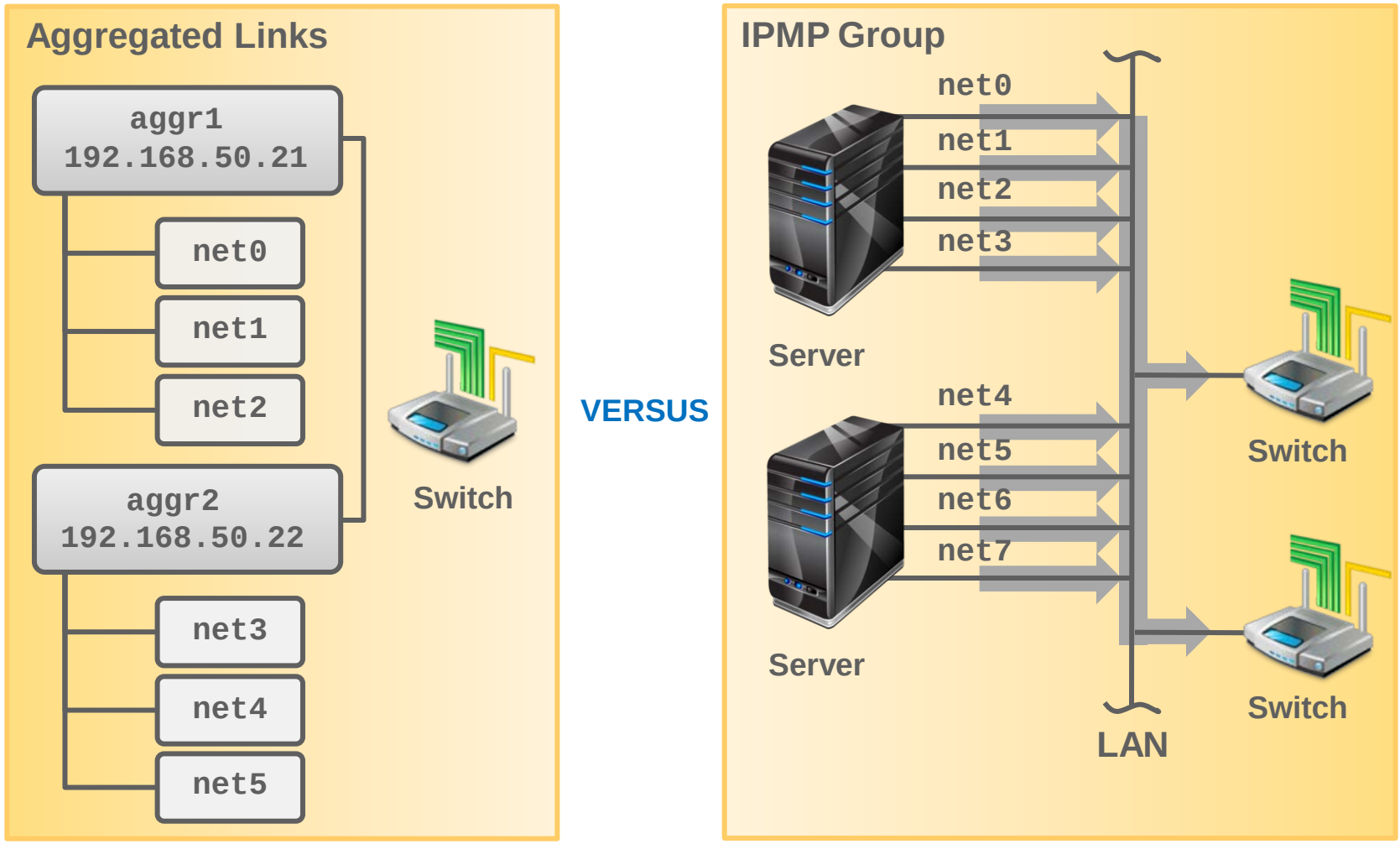
- Performance advantages:
  - Fault tolerance
  - Load spreading
  - Increased bandwidth
  - Transparent redundancy
- IPMP groups:
  - Active-active
  - Active-standby



# IPMP Components

| Component | Description |
|---|---|
| IPMP daemon: `in.mpathd` | Detects interface failures and repairs |
| IPMP service: `svc:/network/ipmp` | Sets IPMP properties, such as enabling or disabling transitive probing |
| Configuration file: `/etc/default/mpathd` | Defines the daemon's behavior |
| IPMP administration command: `ipadm` | Configures IP network interfaces that are part of an IPMP group |
| IPMP display information command: `ipmpstat` | Provides information about the status of IPMP as a whole |
| IP kernel module | Manages outbound load spreading |

# Comparing Link Aggregation and IPMP



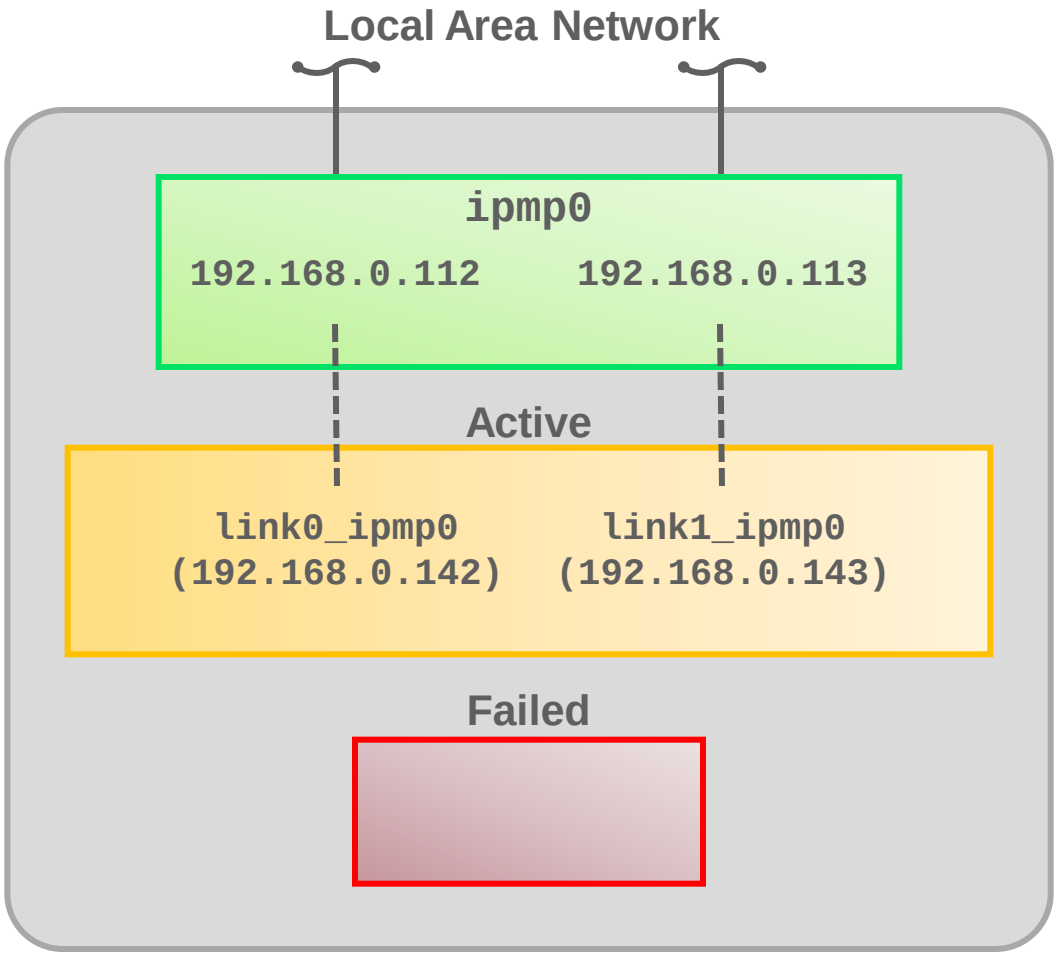
# Failure and Repair Detection in IPMP

- To ensure continuous availability, IPMP performs failure detection on the IPMP group's underlying IP interfaces.
- Failed interfaces remain unusable until they are repaired.
- The remaining active interfaces continue to function, while any existing standby interfaces are deployed as needed.
- The `in.mpathd` daemon handles the following types of failure detection:
  - Probe-based failure detection
    - No test addresses are configured.
      - ICMP probes
      - Transitive probes
    - Test addresses are configured.
  - Link-based failure detection

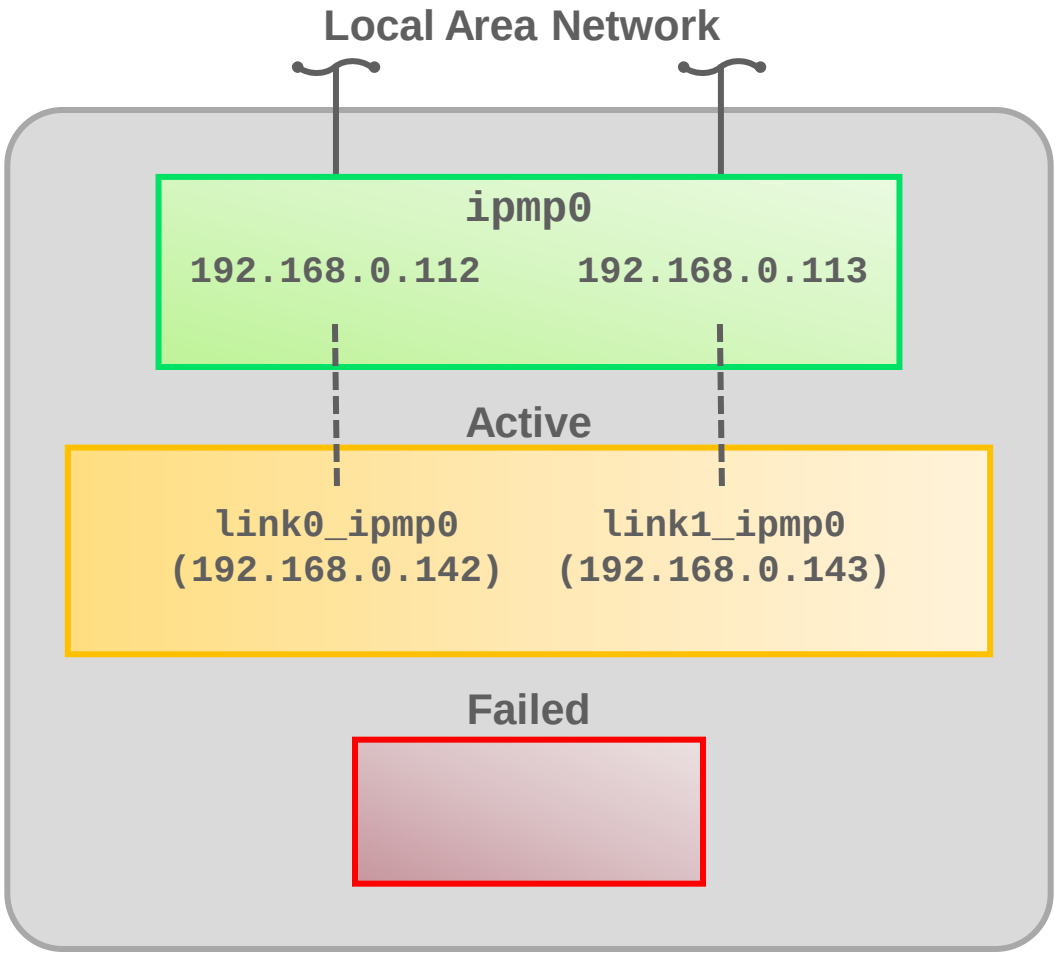
# IPMP Configurations

- An IPMP configuration consists of two or more physical interfaces on the same system that are attached to the same network.
- These interfaces can belong to an IPMP group in either of the following configurations:
  - Active-active
  - Active-standby

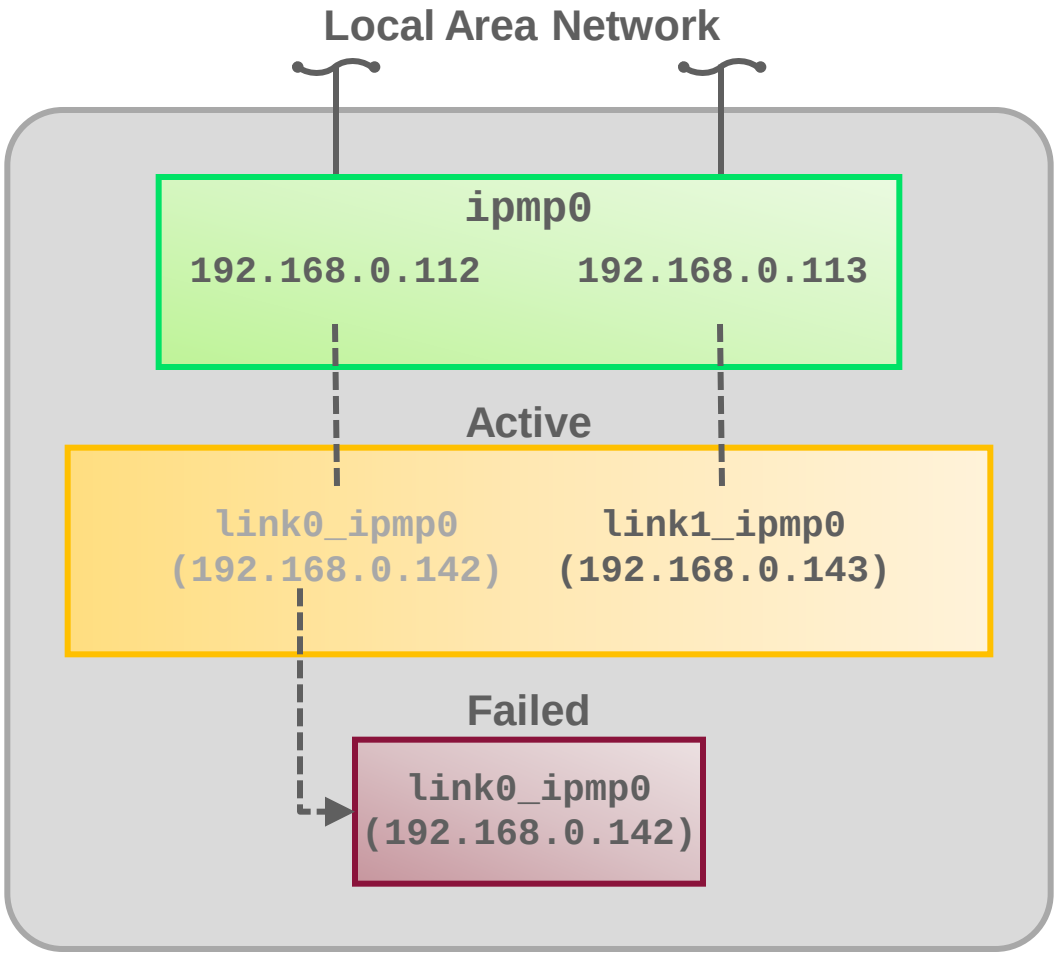
# How IPMP Works: Active-Active



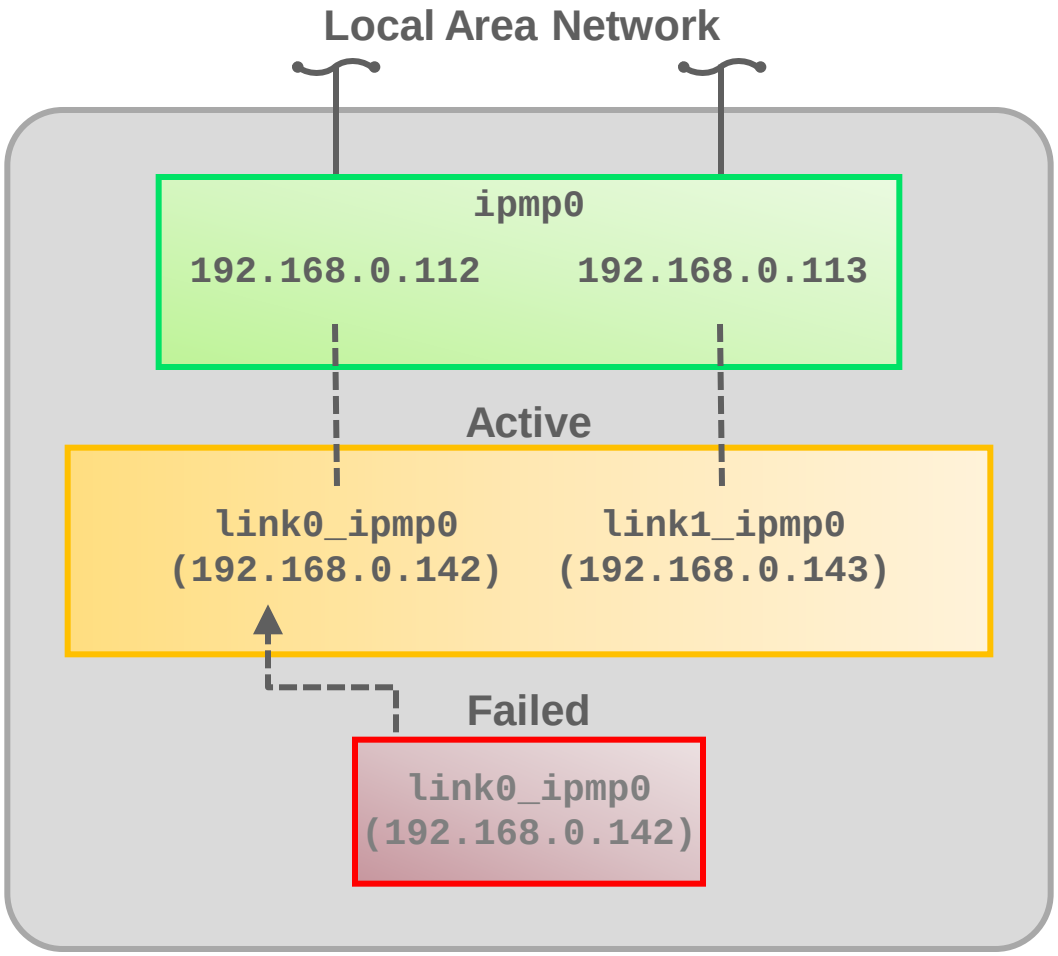
# How IPMP Works: Active-Active



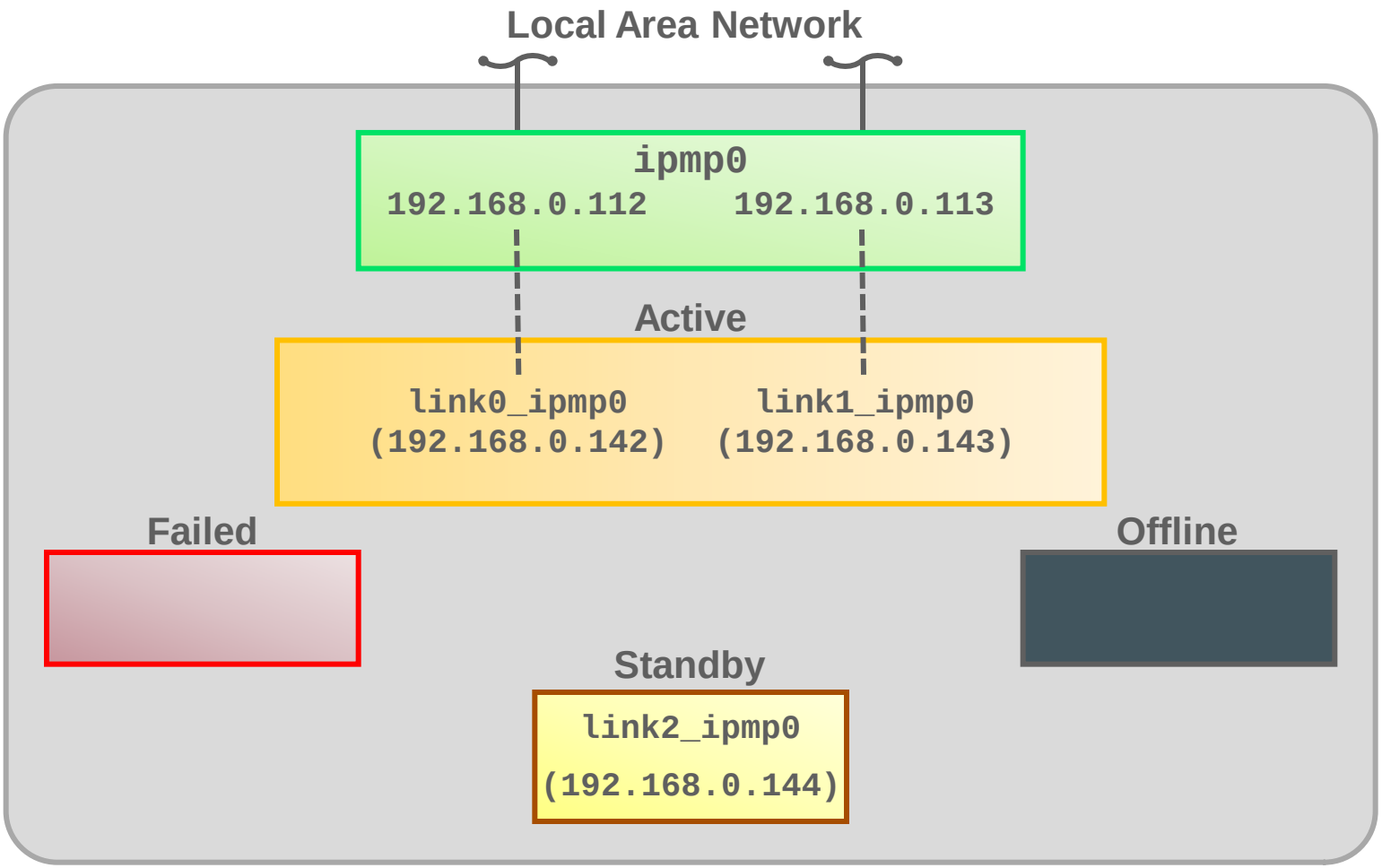
# How IPMP Works: Active-Active



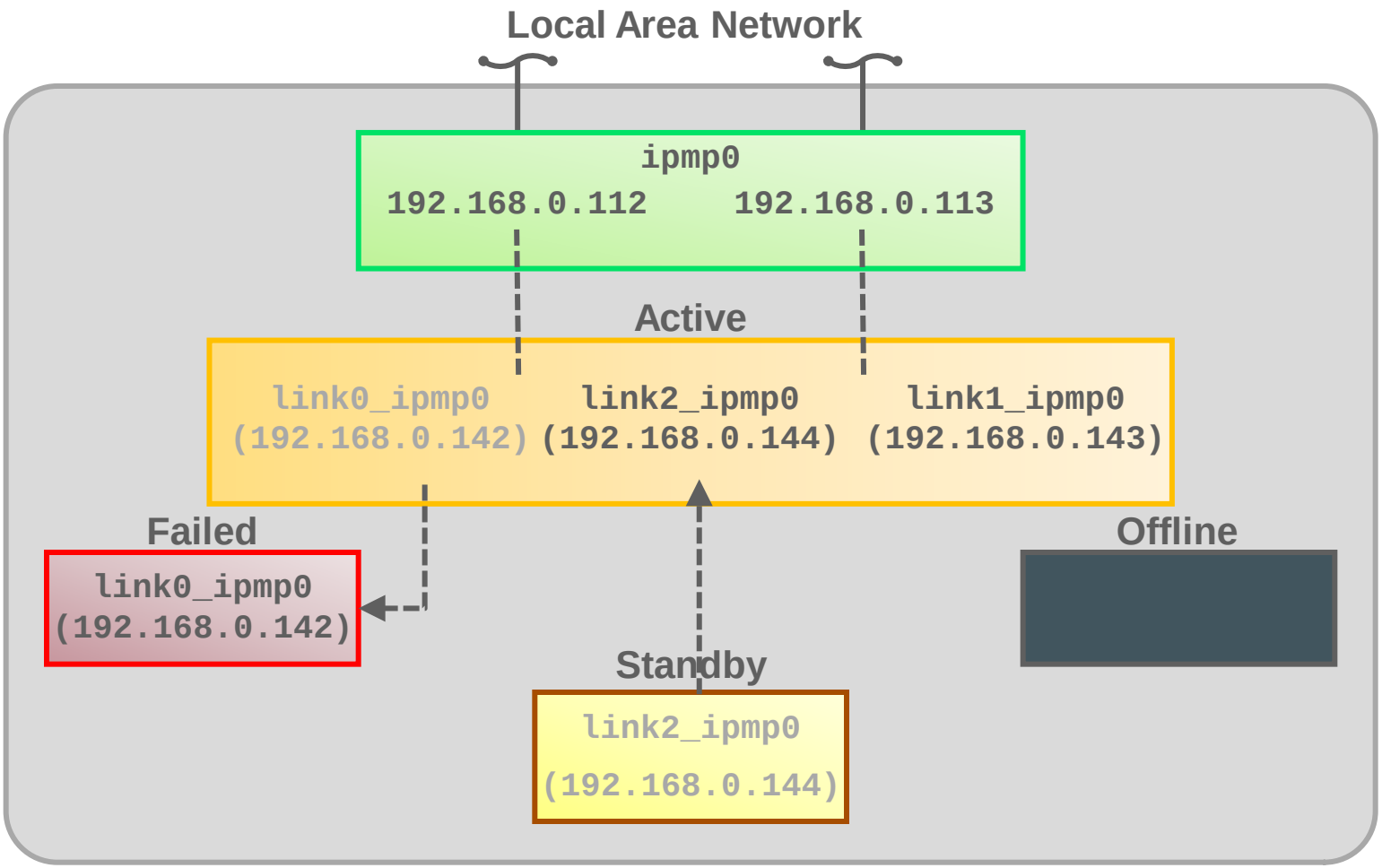
# How IPMP Works: Active-Active



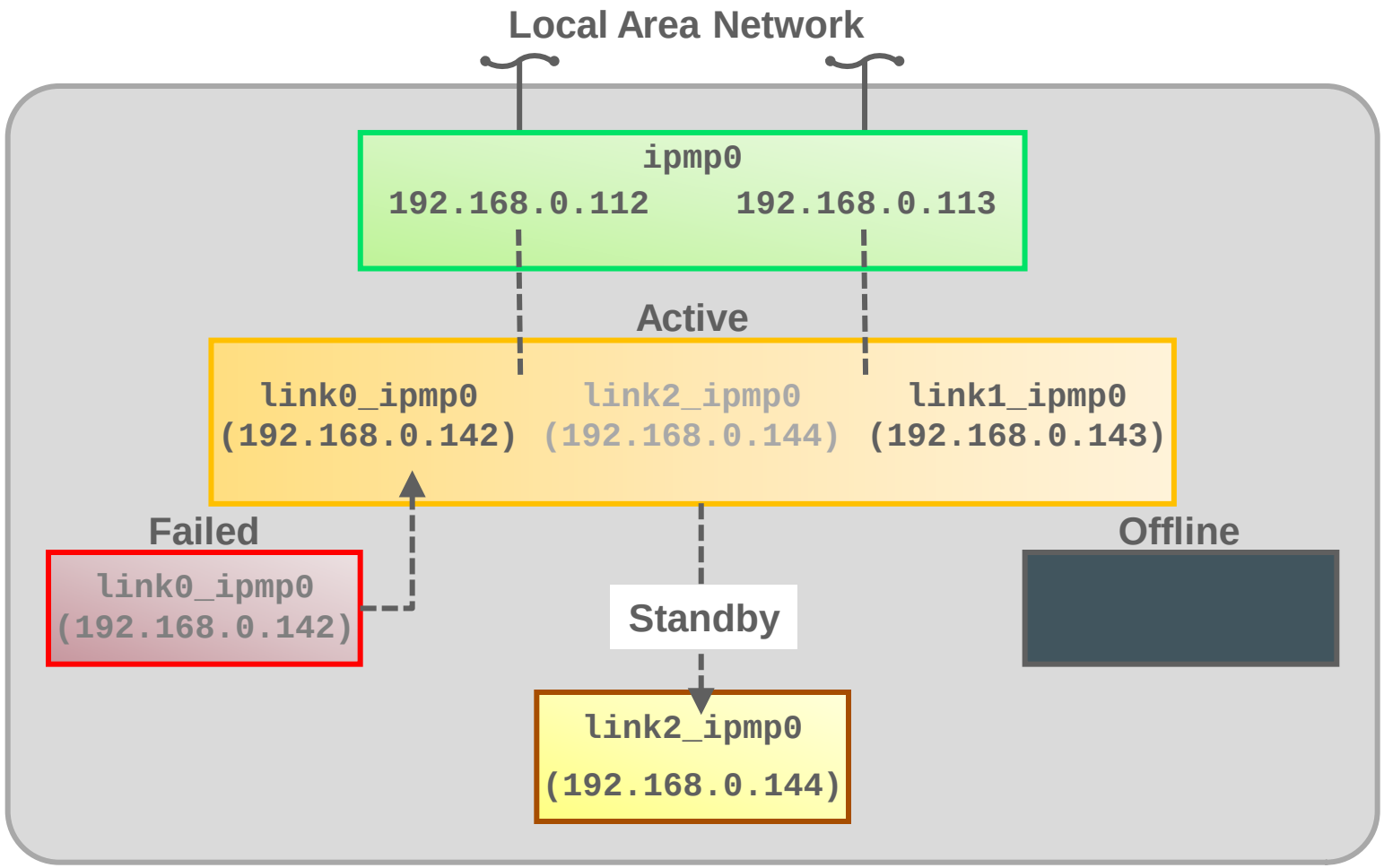
# How IPMP Works: Active-Standby



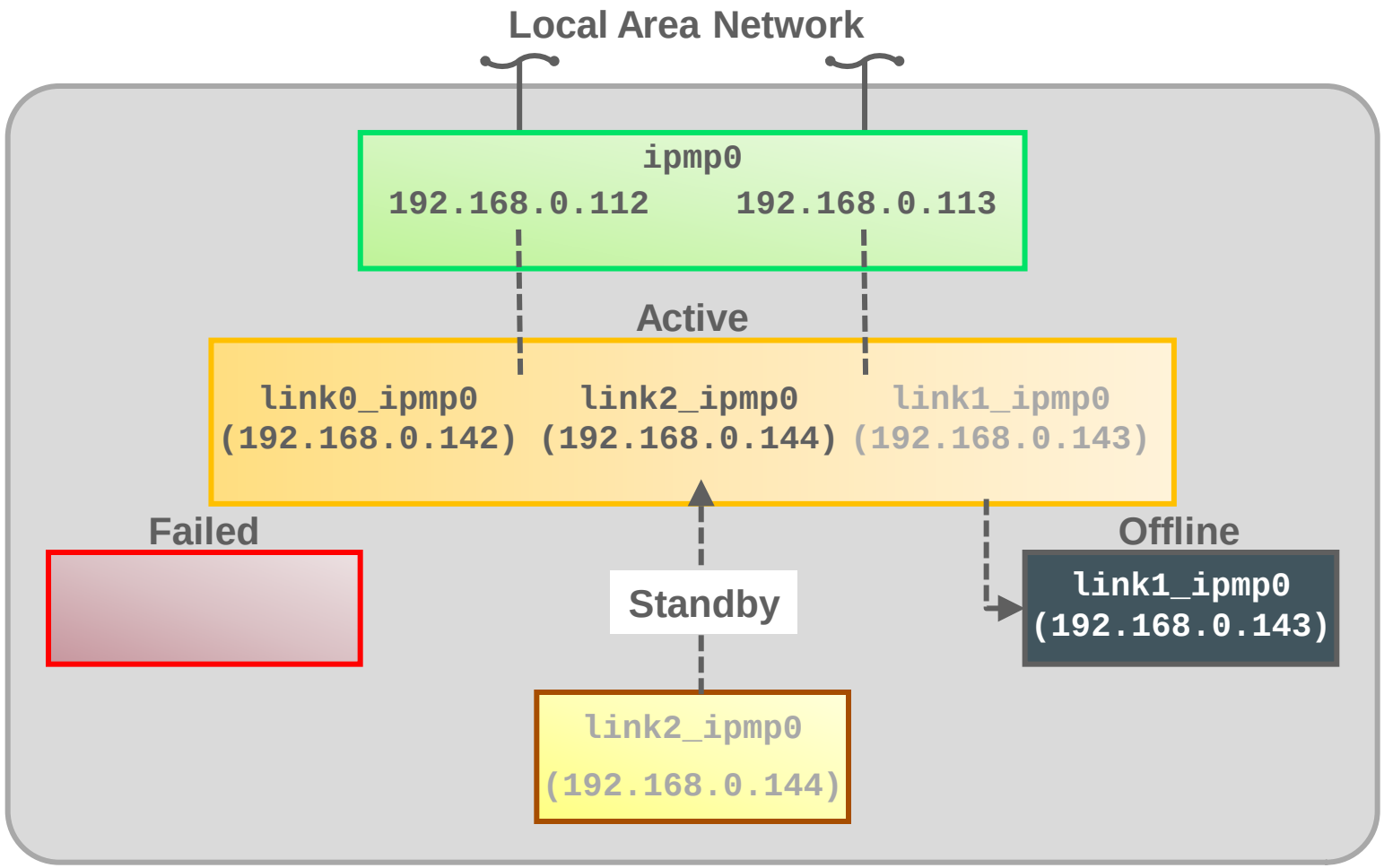
# How IPMP Works: Active-Standby



# How IPMP Works: Active-Standby



# How IPMP Works: Active-Standby



# Configuring an IPMP Group

This section covers the following topics:

- Creating an IPMP group
- Adding IP addresses to an IPMP group
- Moving an interface from one IPMP group to another
- Deleting or disabling an IPMP group

# Creating an IPMP Group

1. Create IP interfaces for the datalinks to use in the IPMP group by using the `ipadm create-ip` command.
2. Create the IPMP group by using the `ipadm create-ipmp` command.

```
# dladm rename-link net0 link0_ipmp0
# dladm rename-link net1 link1_ipmp0
# ipadm create-ip link0_ipmp0
# ipadm create-ip link1_ipmp0
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i link0_ipmp0 -i link1_ipmp0 ipmp0
# ipmpstat -g
GROUP     GROUPNAME  STATE   FDT      INTERFACES
ipmp0     ipmp0      ok      --       link1_ipmp0 link0_ipmp0
```

# Adding IP Addresses to an IPMP Group

1. Add addresses to an IPMP group by using the `ipadm create-addr` command.
2. Verify the results with the `ipadm show-addr` command.

```
# ipadm create-addr -T static -a 192.168.0.112/24 ipmp0/v4add1
# ipadm create-addr -T static -a 192.168.0.113/24 ipmp0/v4add2
# ipadm show-addr
ADDROBJ           TYPE     STATE     ADDR
ipmp0/v4add1      static   ok        192.168.0.112/24
ipmp0/v4add2      static   ok        192.168.0.113/24
```

# Moving an Interface from One IPMP Group to Another Group

1. Remove the interface from the IPMP group by using the `ipadm remove-ipmp` command.
2. Add the interface to another group by using the `ipadm add-ipmp` command.

```
# ipadm remove-ipmp -i link0_ipmp0 ipmp0
# ipadm add-ipmp -i link0_ipmp0 ipmp1
```

# Deleting and Disabling an IPMP Group

To delete an IPMP group, use the `ipadm delete-ip` command.

```
# ipadm delete-ipmp ipmp0
```

To disable an IPMP group, use the `ipadm disable-if` command.

```
# ipadm disable-if -t ipmp0
```

# Implementing Link Failover by Using IPMP

This section covers the configuration of:

- An active-active IPMP group
- An active-standby IPMP group

# Configuring an Active-Active IPMP Group

1. Create IP interfaces by using `ipadm`.
2. Create an IPMP group and add the interfaces to the group.
3. Create static IP addresses for data access.

```
# dladm rename-link net0 link0_ipmp0
# dladm rename-link net1 link1_ipmp0
# ipadm create-ip link0_ipmp0
# ipadm create-ip link1_ipmp0
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i link0_ipmp0 -i link1_ipmp0 ipmp0
# ipadm create-addr -T static -a 192.168.0.112/24 ipmp0/v4add1
# ipadm create-addr -T static -a 192.168.0.113/24 ipmp0/v4add2
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
ipmp0/v4add1      static   ok           192.168.0.112/24
ipmp0/v4add2      static   ok           192.168.0.113/24
lo0/v6            static   ok           ::1/128
```

# Assigning Test Addresses

To assign test addresses to an IPMP subinterface, use `ipadm create-addr -T static -a IP_address link/test`.

```
# ipadm create-addr -T static -a 192.168.0.142/24 link0_ipmp0/test
# ipadm create-addr -T static -a 192.168.0.143/24 link1_ipmp0/test
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
link0_ipmp0/test  static   ok           192.168.0.142/24
link1_ipmp0/test  static   ok           192.168.0.143/24
ipmp0/v4add1      static   ok           192.168.0.112/24
ipmp0/v4add2      static   ok           192.168.0.113/24
lo0/v6            static   ok           ::1/128
```

# Configuring an Active-Standby IPMP Group

1. Set at least one interface's property to `standby` by using the `ipadm set-ifprop` command.
2. Confirm the results.

```
# ipadm show-ifprop -p standby link2_ipmp0
IFNAME      PROPERTY  PROTO PERM  CURRENT   PERSISTENT  DEFAULT POSSIBLE
link2_ipmp0 standby   ip    rw    off       --          off     on,off
# ipadm set-ifprop -p standby=on -m ip link2_ipmp0
# ipadm show-ifprop -p standby link2_ipmp0
IFNAME      PROPERTY  PROTO PERM  CURRENT   PERSISTENT  DEFAULT POSSIBLE
link2_ipmp0 standby   ip    rw    on        on          off     on,off
```

# Monitoring an IPMP Group

This section covers the following topics:

- Displaying IPMP group information
- Obtaining IPMP address information
- Verifying IPMP interface information
- Obtaining probe target information
- Checking probe information

# Displaying IPMP Group Information

To display IPMP group information, use `ipmpstat -g`.

```
# ipmpstat -g
GROUP GROUPNAME STATE FDT    INTERFACES
ipmp0     ok    10.00s link1_ipmp0 link0_ipmp0 (link2_ipmp0)
```

# Obtaining IPMP Address Information

To display IPMP address information, use `ipmpstat -an`.

```
# ipmpstat -an
ADDRESS         STATE  GROUP  INBOUND     OUTBOUND
::              down   ipmp0  --          --
192.168.0.113   up     ipmp0  link1_ipmp0 link0_ipmp0
192.168.0.112   up     ipmp0  link0_ipmp0 link1_ipmp0 link0_ipmp0
```

# Verifying IPMP Interface Information

To verify IPMP interface information, use `ipmpstat -i`.

```
# ipmpstat -i
INTERFACE    ACTIVE  GROUP      FLAGS     LINK    PROBE    STATE
link2_ipmp0  yes     ipmp0      -s-----   up      ok
link1_ipmp0  yes     ipmp0      --mbM--   up      ok
link0_ipmp0  no      ipmp0      -------   up      failed
```

# Obtaining Probe Target Information

To display information about test address targets, use `ipmpstat -nt`.

```
# ipmpstat -nt
INTERFACE   MODE       TESTADDR           TARGETS
link1_ipmp0 multicast  192.168.0.143      192.168.0.100 192.168.0.111
link0_ipmp0 multicast  192.168.0.142      192.168.0.100 192.168.0.111
```

# Checking Probe Information

To check probe information, use `ipmpstat -pn`.

```
# ipmpstat -pn
TIME      INTERFACE   PROBE  NETRTT  RTT       RTTAVG    TARGET
0.06s     link2_ipmp0 i163   0.26ms  0.49ms    0.33ms    192.168.0.100
0.90s     link1_ipmp0 i162   0.26ms  0.39ms    0.31ms    192.168.0.100
0.92s     link2_ipmp0 i164   0.19ms  0.36ms    0.34ms    192.168.0.100
0.49s     link0_ipmp0 i161   --      --        --        192.168.0.100
-0.49s    link0_ipmp0 i160   --      --        --        192.168.0.100
2.52s     link2_ipmp0 i165   0.23ms  0.39ms    0.34ms    192.168.0.100
2.74s     link1_ipmp0 i163   0.24ms  0.38ms    0.32ms    192.168.0.100
3.69s     link1_ipmp0 i164   0.25ms  0.45ms    0.34ms    192.168.0.100
2.31s     link0_ipmp0 i162   --      --        --        192.168.0.100
…
…
…
<Ctrl+C>
```

# Quiz

What is the default policy for link aggregation?

a. L2 (Networking): MAC header
b. L3 (Addressing): IP header
c. L4 (Communication): TCP/UDP or other ULP header

# Quiz



IPMP can be configured for both IPv4 and IPv6.

a. True

b. False

# Quiz

Which IPMP component is responsible for detecting failures?

a. IPMP daemon
b. IPMP service
c. DHCP

# Quiz



Link aggregation and IPMP cannot be deployed together.

a. True

b. False

# Practice 5-3 Overview: Configuring IPMP

This practice covers the following topics:

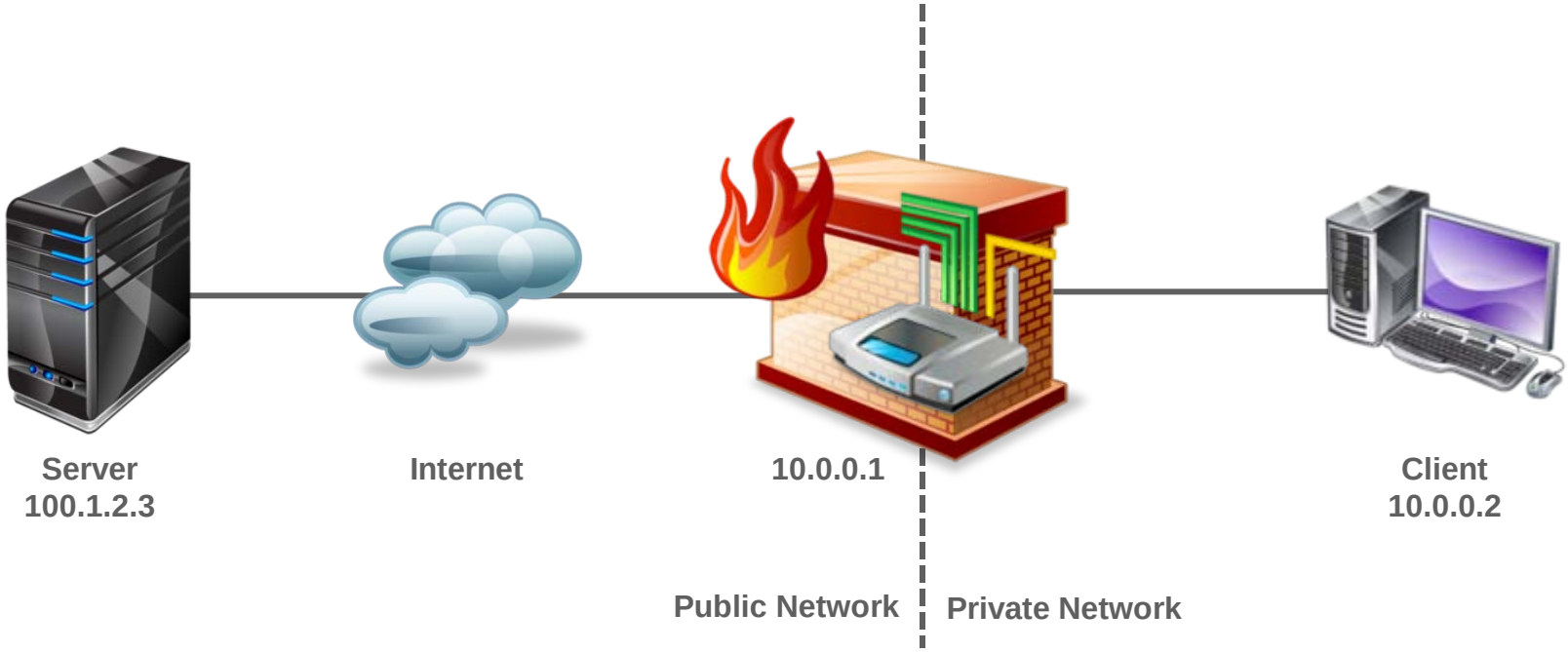
- Configuring an active-active IPMP configuration
- Configuring an active-standby IPMP configuration
- Removing the IPMP configuration

# Agenda

- Configuring virtual switches
- Configuring link aggregation for high performance
- Configuring IPMP for IP high availability
- Configuring Packet Filter to control network access

# Need for a Firewall

A firewall is a facility that restricts access between a protected network and an unprotected network (such as the Internet) or between other sets of networks based on the security policy of the organization.



# Packet Filter: Overview

- The Packet Filter (PF) feature of Oracle Solaris is a network firewall that:
  - Is based on OpenBSD PF version 5.5
  - Captures incoming packets and evaluates them for entry to and exit from the system
  - Provides stateful packet inspection
  - Matches packets by IP address and port number as well as by the receiving network interface
- Both PF and IP Filter features are available for filtering packets in Oracle Solaris 11.3.
- **Note:** Because PF is a more robust filtering module, you should transfer your firewall policy from IP Filter rules to PF.

# Comparison of IP Filter and PF

| Firewall Feature | IP Filter | PF |
|---|---|---|
| **Configuration files** | Several, such as `ippool.conf`, `ipnat.conf`, and `ipv6.conf` | One `pf.conf` file |
| **Package name** | `ipfilter` | `firewall`, not installed by default |
| **pass rules** | Stateless by default | Stateful by default |
| **Rights profile** | Network Security | Network Firewall Management |
| **SMF service name** | `ipfilter`, enabled by default | `firewall` |
| **IPv4 and IPv6 packet fragments** | IP reassembly must be explicitly turned on | IP reassembly is on by default |
| **Loopback interface protection** | Must be enabled by `set intercept_lopback true;` | Firewall always intercepts packets on loopback interface |
| **OS signature file** | None | `pf.os` |

# Behavior of PF Firewall

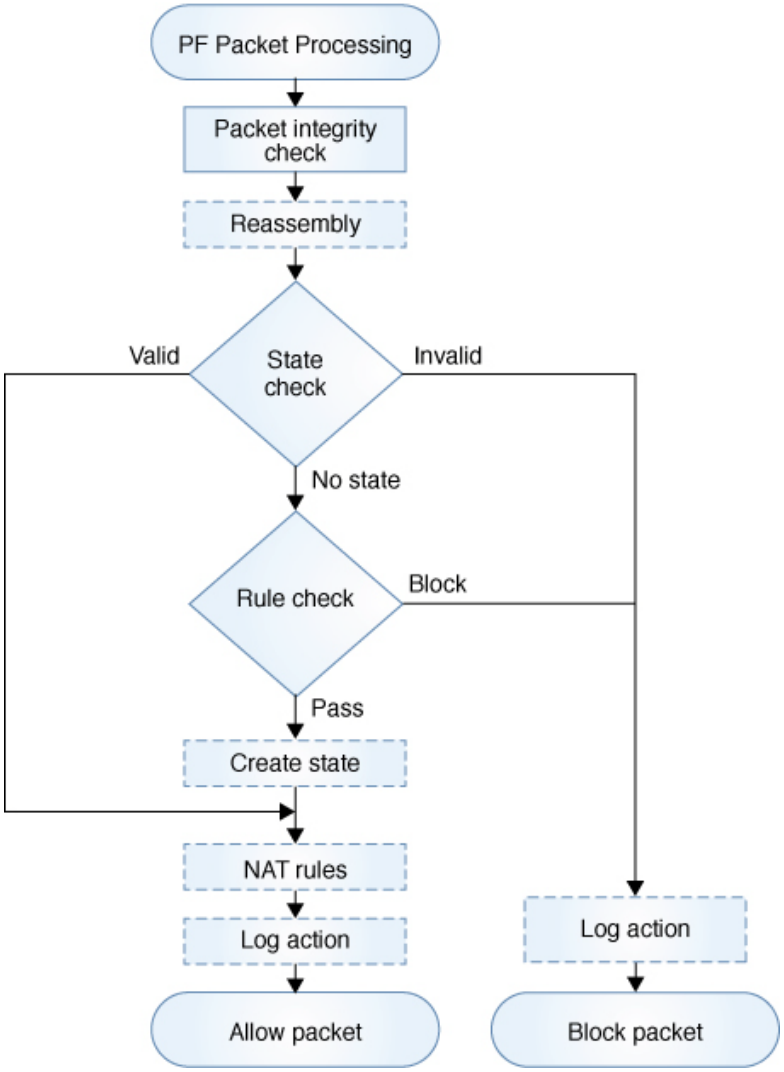
- The PF firewall is controlled by the `svc:/network/firewall` SMF service that loads the rules from the `/etc/firewall/pf.conf` configuration file.
- A rule helps process packets and determine whether they are accepted or dropped.
- You can use the `pfconf` script to edit the PF configuration file.
- When you save the file, the script verifies the syntax of the firewall rules and then refreshes the firewall service to put the new rules in effect.

# PF Rules

- A rule in the `/etc/firewall/pf.conf` configuration file contains the following parts:
  - Actions: Define the action that applies to the packet if the packet matches the rule.
  - Match parameters: Define criteria that determine whether a packet matches the rule.
  - Optional actions: Define additional optional actions.
- You write a rule by using the following elements in order:
  - Begin the rule with an action.
  - Match desired parameters.
  - Include desired optional actions.
- PF rule syntax:

```
# action match-parameter optional-action-1 optional-action-2…
```

# Packet Flow in the PF Firewall



# Configuring PF Firewall

1. Install the PF package.

```
# pkg install firewall
```

2. Create or update the packet filtering rule set.

```
# pfconf
```

The `pfconf` script uses the service property rules for the location of the PF configuration file.

3. Disable the `ipfilter` service first, then enable the PF.

```
# svcadm disable network/ipfilter
# svcadm enable network/firewall
```

4. (Optional) Disable the PF service.

```
# svcadm disable network/firewall
```

This command removes all rules from the kernel and disables the PF service.

# Monitoring PF Firewall

- Examine the status of the `firewall` service.

```
# svcs -x firewall:default
svc:/network/firewall:default (Network Firewall)
 State: disabled since Fri Apr 10 10:10:50 2015
Reason: Disabled by an administrator.
   See: http://oracle.com/msg/SMF-8000-05
   See: pf.conf(5)
   See: /var/svc/log/network-firewall:default.log
Impact: This service is not running.
```

- List the configuration file names and locations for PF service.

```
# svccfg -s firewall:default listprop | grep firewall
firewall                      application
firewall/fingerprints         astring     /etc/firewall/pf.os
firewall/rules                astring     /etc/firewall/pf.conf
firewall/value_authorization  astring     solaris.smf.value.network.firewall
restarter/logfile             astring     /var/svc/log/network-firewall:default.log
```

# Monitoring PF Firewall

- Examine the current rules of the PF firewall.

```
# pfctl -s rules
empty list for firewall(out)
pass in quick on net1 from 192.168.1.0/24 to any
pass in all
block in on net1 from 192.168.1.10/32 to any
```

- Verify the PF firewall configuration.

```
# pfctl -n -f /test/firewall/pf.conf
```

# Summary

In this lesson, you should have learned how to configure:

- A virtual switch
- Link aggregation for high performance
- IPMP for IP high availability
- Packet Filter to control network access