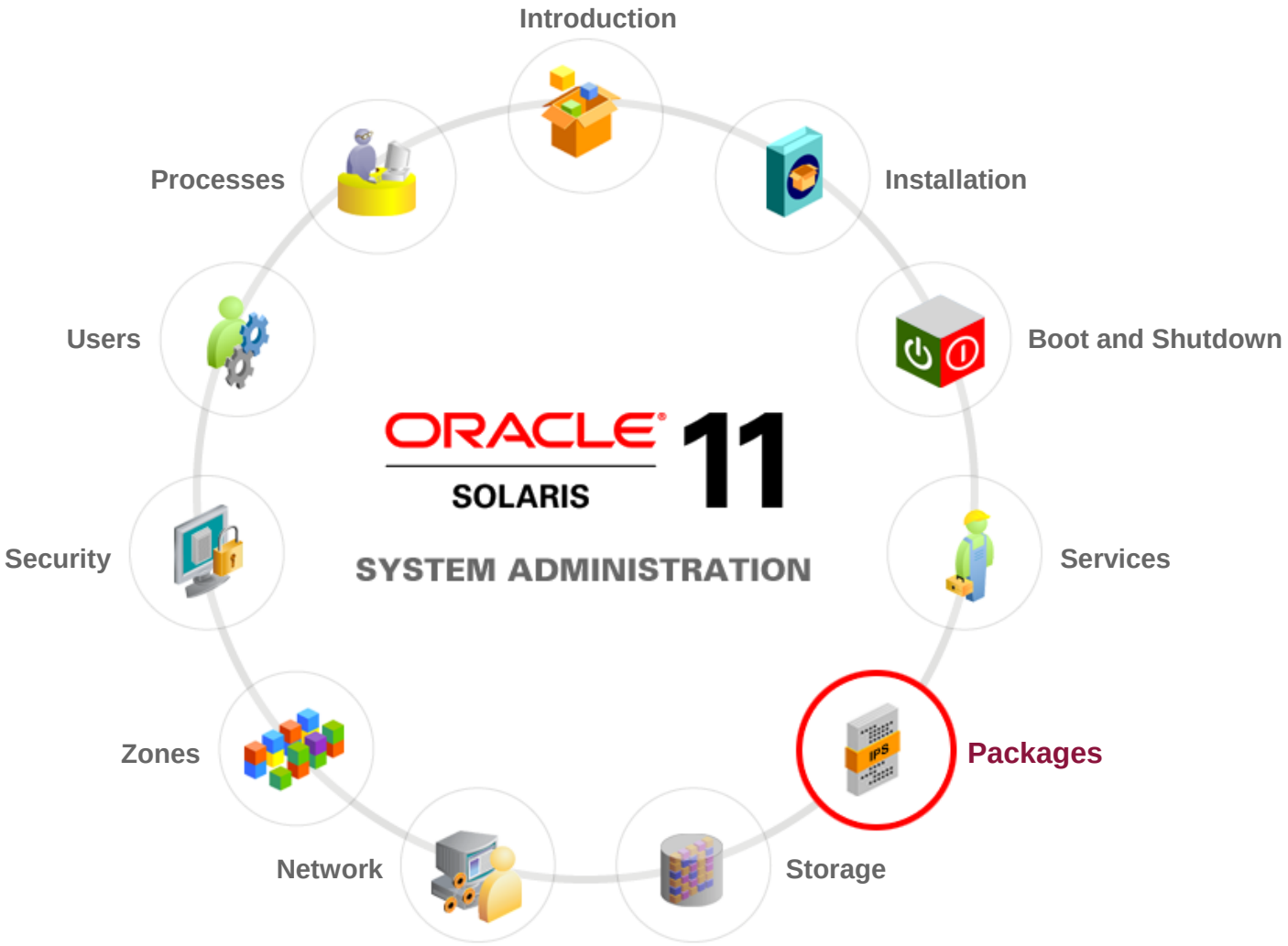
5

Administering Software Packages by Using IPS and Managing Boot Environments

Workflow Orientation



Objectives

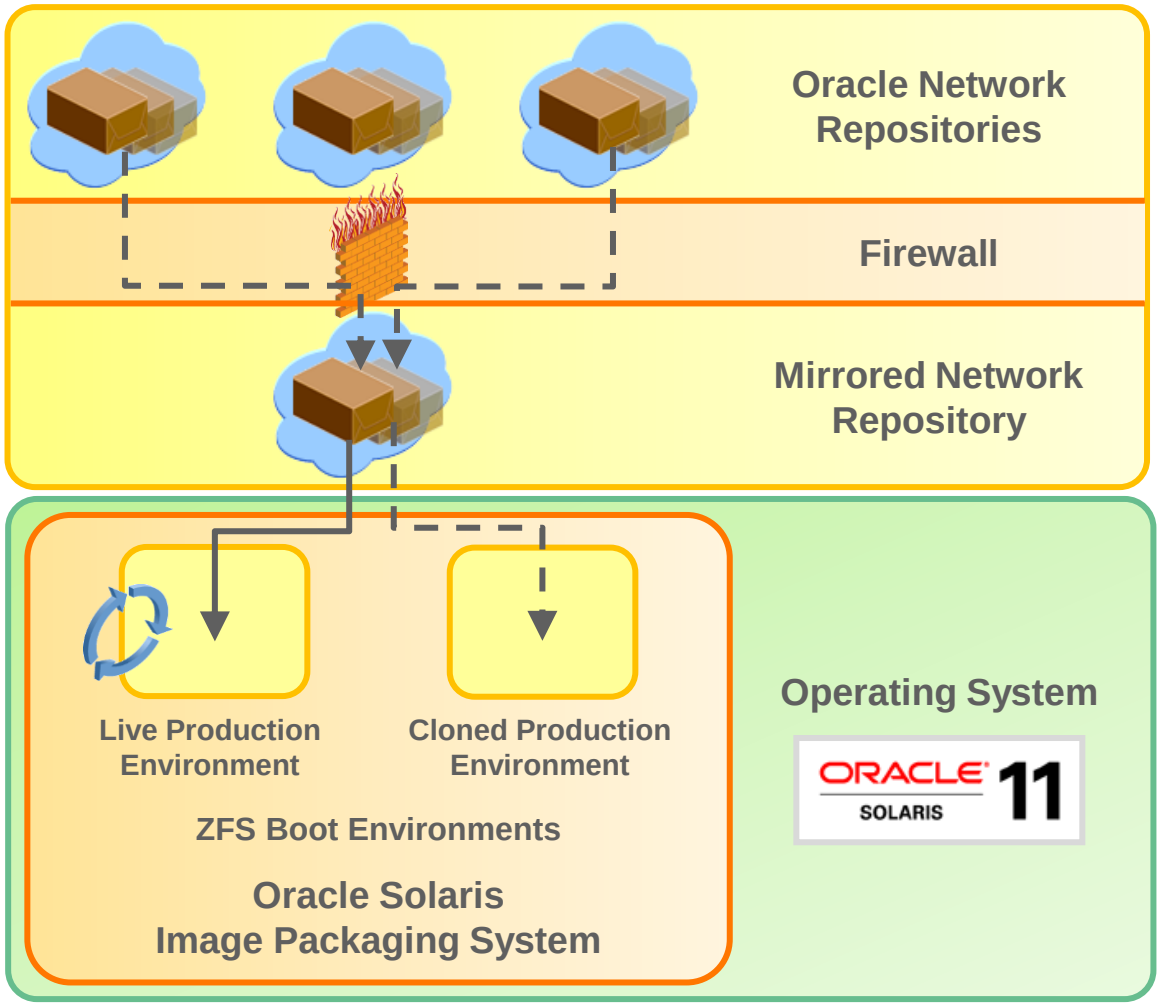
After completing this lesson, you should be able to:

- Describe IPS, its components, and interfaces
- Configure an IPS client to access the local IPS repository
- Manage package publishers
- Manage software packages
- Manage signed packages and package properties
- Describe the use of IPS in packaging the Oracle Solaris OS
- Manage boot environments
- Update the OS by using IPS

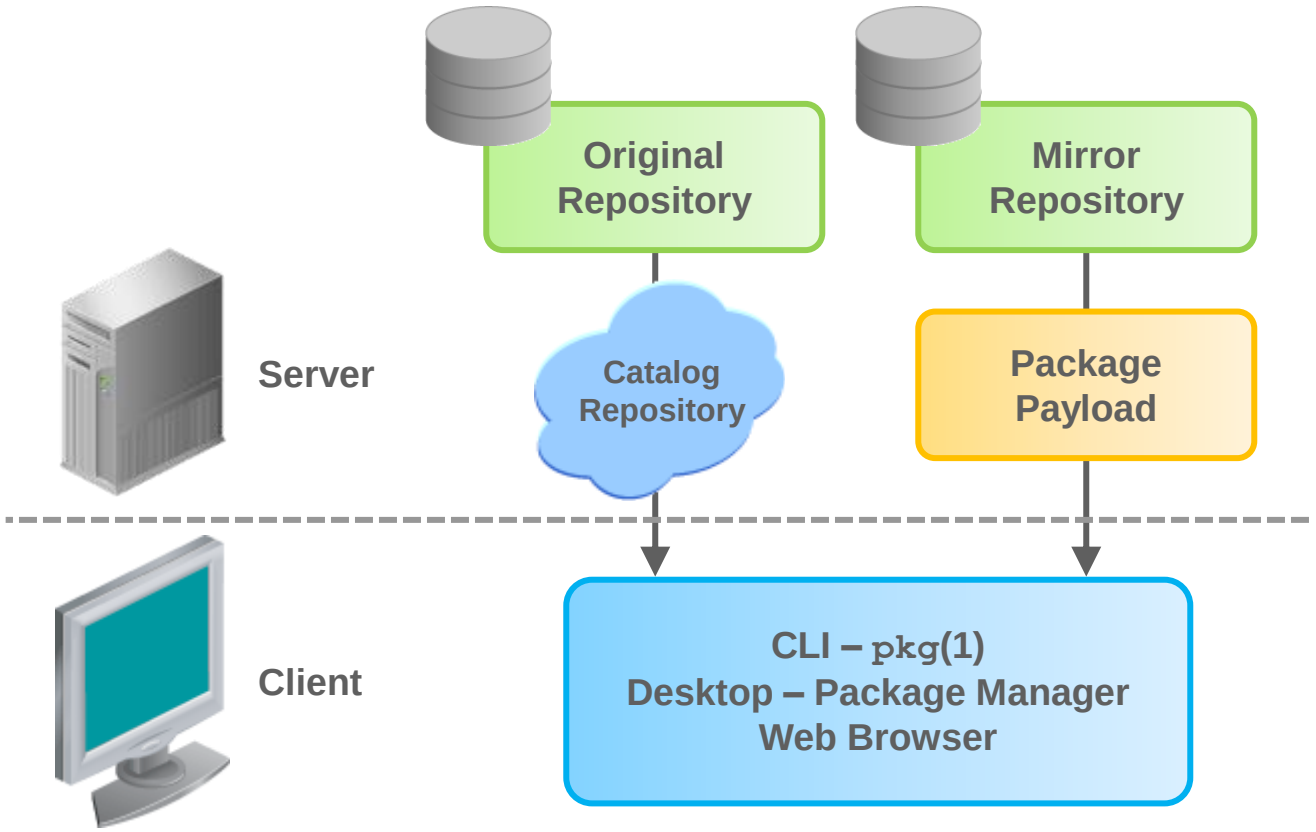
Agenda

- **Describing IPS, Its Components, and Interfaces**
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- Managing Software Packages
- Managing Signed Packages and Package Properties
- Describing the use of IPS in packaging the Oracle Solaris OS
- Managing Boot Environments
- Updating the OS by using IPS

Introducing IPS



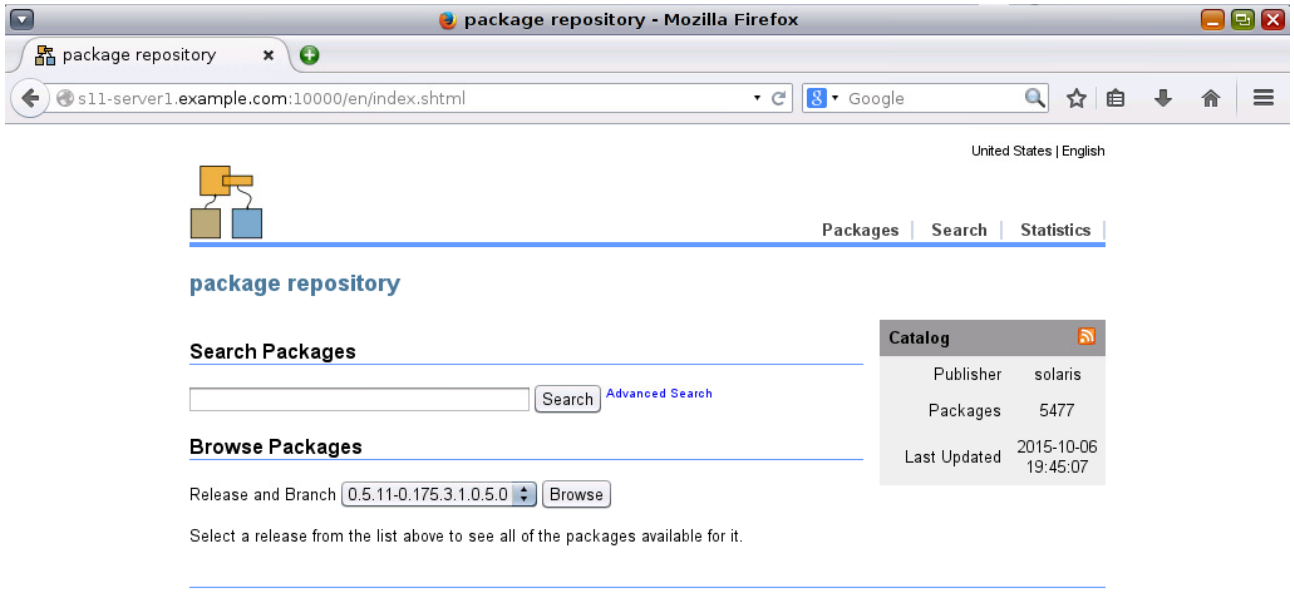
Introducing IPS Components



Introducing the IPS Interfaces

IPS supports the following interfaces:

- Command line
- Browser User Interface (BUI)



Package Dependencies

- Dependencies define how packages are related.
- All dependencies must be satisfied for a successful package installation.
- A package can depend on another package because the first requires functionality in the second for the functionality in the first to work or to install.
- If a dependency fails during installation, the packaging system attempts to install or update the dependent package to a sufficiently new version, subject to other constraints.
- Dependencies can be optional.

Package Dependency Types

Type	Description
require	The target package is required and must have a version that is equal to or later than the version specified in the source package. A package cannot be installed if any of its require dependencies are not satisfied.
require-any	Any one of the multiple target packages, as specified in the source package, can satisfy the dependency by following the same rules as the `require` dependency type.
optional	The dependency target, if present, must be at the specified version level or later.
conditional	The dependency target is required only if the package defined by the predicate attribute is present in the system.
group	The dependency target is required unless the package is in the image avoid list.

Package Dependency Types

Type	Description
origin	Before installation of this package, the dependency target must, if present, be at the specified value or at a greater value in the image to be modified.
incorporate	The dependency is optional, but the version of the target package is constrained.
parent	The dependency is ignored if the image is not a child image, such as a zone. If the image is a child image, the dependency target must be present in the parent image.
exclude	The containing package cannot be installed if the dependency target is present at the specified version level or at a later version level. If no version is specified, the target package cannot be installed concurrently with the package that specifies the dependency.

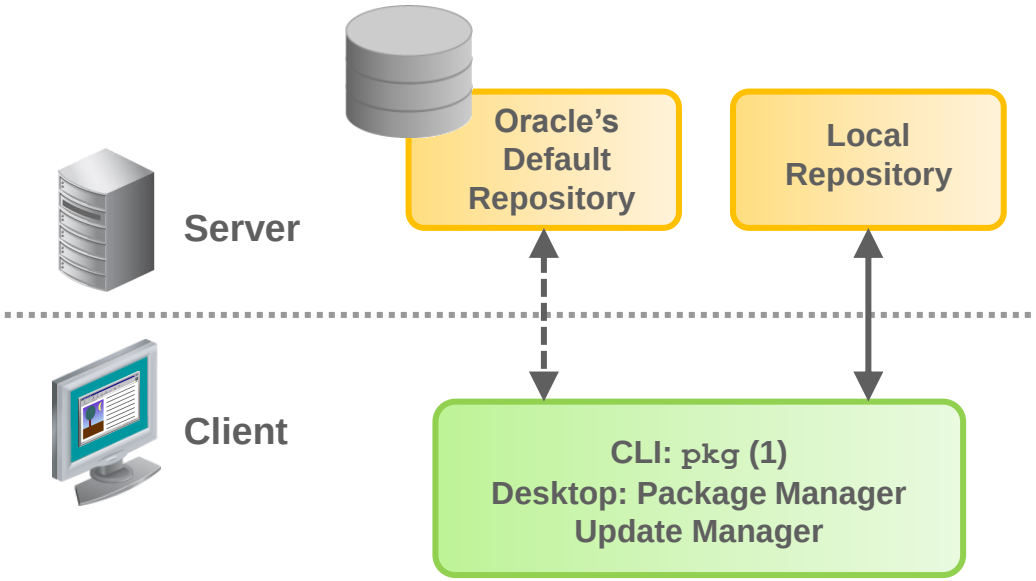
Agenda

- Describing IPS, Its Components, and Interfaces
- **Configuring an IPS Client to Access the Local IPS Repository**
- Managing Package Publishers
- Managing Software Packages
- Managing Signed Packages and Package Properties
- Describing the use of IPS in packaging the Oracle Solaris OS
- Managing Boot Environments
- Updating the OS by using IPS

Importance of IPS and Package Management

A local IPS repository provides the following benefits:

- Performance and security
- Replication
- Customized packages



Configuring an IPS Client to Access the Local IPS Repository

Required tasks:

1. Determining the client host and domain names
2. Checking network connectivity
3. Setting the publisher
4. Testing client access to the local IPS server

Determining the Client Host and Domain Names

Use the `hostname` and `domainname` commands to identify the client machine.

```
# hostname
s11-desktop

# domainname
example.com
```

Checking Network Connectivity

Verify DNS service access and connectivity with the local IPS server.

```
# nslookup s11-server1
Server:         192.168.1.100
Address:192.168.1.100#53

Name:           s11-server1.example.com
Address:192.168.1.100

# ping s11-server1
s11-server1 is alive
```

Setting the Publisher

Use the `pkg set-publisher` command to set the publisher to point to the local IPS repository.

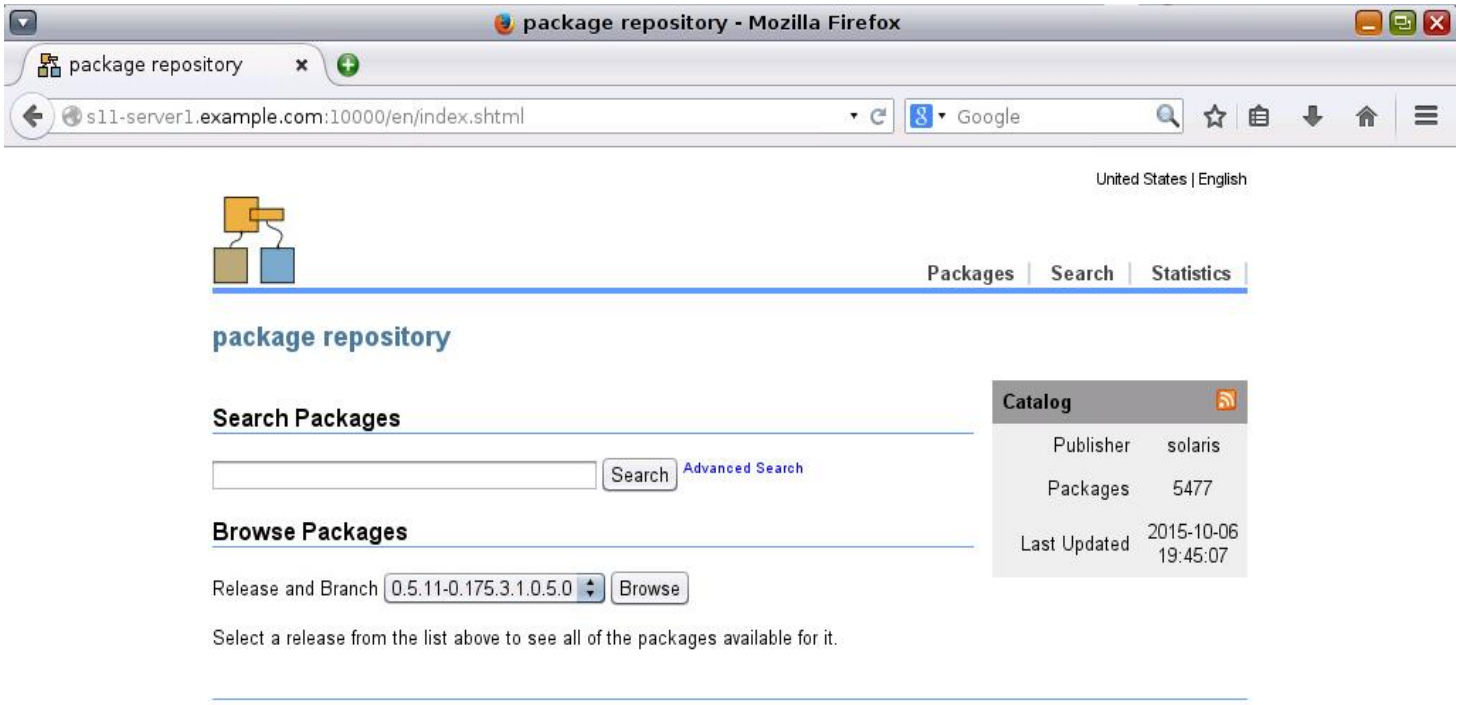
```
# pkg publisher
PUBLISHER    TYPE         STATUS P LOCATION
solaris      origin       online F http://pkg.oracle.com/solaris/release

# pkg set-publisher -G '*' -g http://s11-server1.example.com/ solaris

# pkg publisher
PUBLISHER    TYPE         STATUS P LOCATION
solaris      origin       online F http://s11-server1.example.com:10000
```

Testing Client Access to the Local IPS Server

To test client access to the IPS server, open the local publisher URI in a browser.



Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- **Managing Package Publishers**
- Managing Software Packages
- Managing Signed Packages and Package Properties
- Describing the use of IPS in packaging the Oracle Solaris OS
- Managing Boot Environments
- Updating the OS by using IPS

Managing Package Publishers

This section covers the following topics:

- Displaying publisher information
- Specifying publisher rankings
- Specifying publisher stickiness
- Setting the publisher search order
- Disabling or enabling a publisher
- Changing a publisher’s origin URI

Displaying Publisher Information

- To display only the highest-ranked publisher in the search order, run `pkg publisher -P`.

```
# pkg publisher -P
PUBLISHER                 TYPE     STATUS   P LOCATION
solaris                   origin   online   F http://s11-
server1.example.com:10000/
```

- To display information about a specific publisher, run `pkg publisher` *`publisher_name`*.

```
# pkg publisher solaris
      Publisher: solaris
          Alias:
     Origin URI: http://s11-server1.example.com:10000/
        SSL Key: None
       SSL Cert: None
    Client UUID: 55dc8a86-fbe5-11e3-bc11-811d2d030777
 Catalog Updated: June 25, 2015 03:55:23 AM
        Enabled: Yes
```

Specifying Publisher Rankings

To set a publisher to be the highest-ranked publisher in the search order, run `pkg set-publisher -P` *`publisher_name`* or the `--search-first` option.

```
# pkg publisher
PUBLISHER                 TYPE    STATUS   P LOCATION
solaris                   origin  online   F http://s11-
server1.example.com:10000
whoisit.com (non-sticky) origin  online   F
http://pkg.example.com/release

# pkg set-publisher -P whoisit.com

# pkg publisher
PUBLISHER                 TYPE    STATUS   P LOCATION
whoisit.com               origin  online   F
http://pkg.example.com/release
Solaris (non-sticky)      origin  online   F http://s11-
server1.example.com:10000
```

Specifying Publisher Stickiness

To make a publisher non-sticky, run `pkg set-publisher --non-sticky` *`publisher_name`*.

```
# pkg set-publisher --non-sticky example.com
```

To make a publisher sticky, run `pkg set-publisher --sticky` *`publisher_name`*.

```
# pkg set-publisher --sticky example.com
```

Example: Sticky or Non-sticky

Pub1

- pkg1_v1

Pub2

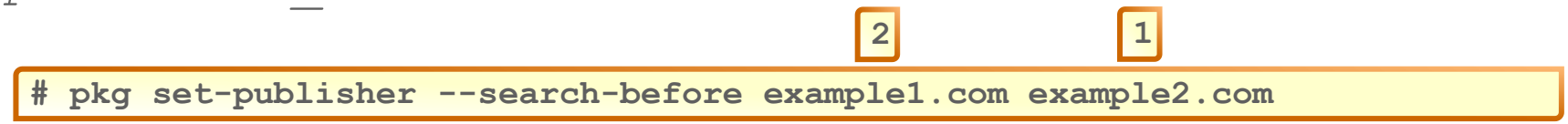
- pkg1_v2

client currently installed with pkg1_v1 from Pub1, then perform pkg update

- if Pub1 set to sticky, pkg1_v1 from Pub1 will not be updated
- else non-sticky, pkg1_v1 from Pub1 will be updated to pkg1_v2 using Pub2

Setting the Publisher Search Order

To move a publisher higher in the search order, run `pkg set-publisher --search-before` *`publisher_name`* *`publisher_name`*.



To move a publisher lower in the search order, run `pkg set-publisher --search-after` *`publisher_name`* *`publisher_name`*.

Disabling and Enabling a Publisher

To disable a publisher, run `pkg set-publisher -d` *`publisher_name`*.

```
# pkg set-publisher -d solaris.com
```

To enable a publisher, run `pkg set-publisher -e` *`publisher_name`*.

```
# pkg set-publisher -e solaris.com
```

NOTE:
- **client cannot update, install, list from the publisher**
- **still can modify properties (configured as proxy/SSL and so on)**

Changing a Publisher's Origin URI

To change a publisher's origin URI, run `pkg set-publisher -g` *`newpublisher_URI -G oldpublisher_URI newpublisher`*.

```
# pkg set-publisher -g http://pkg.example.com/support \
-G http://pkg.example.com/release solaris
```

Quiz

You want to set `mypublisher.com` as the highest-ranked publisher for your local IPS repository. Which command would you use to execute this task?

a. `pkg publisher -P mypublisher.com`
b. `pkg publisher -n mypublisher.com`
c. `pkg set-publisher -P mypublisher.com`
d. `pkg set-publisher -e mypublisher.com`

Quiz

You want to set `mypublisher.com` as the highest-ranked publisher for your local IPS repository. Which command would you use to execute this task?

a. `pkg publisher -P mypublisher.com`

b. `pkg publisher -n mypublisher.com`

c. `pkg set-publisher -P mypublisher.com`

d. `pkg set-publisher -e mypublisher.com`

Quiz

You have three publishers listed in the following order: `mypublisher.com` (the highest-ranked publisher), `solaris`, and `whoisit`. For search order purposes, you want to move the `whoisit` publisher before the `solaris` publisher. Which command would you use to execute this task?

a. `pkg set-publisher --search-before solaris whoisit`

b. `pkg set-publisher --search-after solaris whoisit`

c. `pkg set-publisher --search-before whoisit solaris`

Quiz

You have three publishers listed in the following order: `mypublisher.com` (the highest-ranked publisher), `solaris`, and `whoisit`. For search order purposes, you want to move the `whoisit` publisher before the `solaris` publisher. Which command would you use to execute this task?

a. `pkg set-publisher --search-before solaris whoisit`

b. `pkg set-publisher --search-after solaris whoisit`

c. `pkg set-publisher --search-before whoisit solaris`

Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- **Managing Software Packages**
- Managing Signed Packages and Package Properties
- Describing the use of IPS in packaging the Oracle Solaris OS
- Managing Boot Environments
- Updating the OS by using IPS

Managing Software Packages by Using the CLI

- Listing package state information
- Displaying package information
- Displaying the contents of a package
- Updating and installing packages
- Viewing a package installation action without installing
- Verifying a package installation
- Searching for a package
- Uninstalling a package

Listing Package State Information

To list package state information, run `pkg list` *`pkg-fmri`*.

```
# pkg list entire
NAME (PUBLISHER)        VERSION                         IFO
entire                  0.5.11-0.175.3.0.0.27.0         i--
```

Displaying Package Information

To display package information, run `pkg info` *`pkg-fmri`*.

```
# pkg info -r apptrace
          Name: developer/apptrace
       Summary: Apptrace Utility
   Description: Apptrace utility for application tracing, including
                shared objects
      Category: Development/System
         State: Not installed
     Publisher: solaris
       Version: 0.5.11
 Build Release: 5.11
        Branch: 0.175.3.0.0.27.1
Packaging Date: July 18, 2015 06:39:19 PM
          Size: 164.98 kB
          FMRI: pkg://solaris/developer/apptrace@0.5.11,5.11-
    0.175.3.0.0.27.1:20150717T222030Z
```

Displaying Package Information

To display package information, run `pkg info` –s *pkg-fmri*.

```
# pkgrepo info -s http://pkg.oracle.com/solaris/release/

PUBLISHER    PACKAGES    STATUS    UPDATED
solaris      6143        online    2018-04-15T00:45:52.227891Z
```

Understanding Package FMRI (Fault Management Resource Indicator)

```
pkg://solaris/developer/gcc@5.4.0,5.11-
0.175.3.22.0.1.0:20170620T160441Z
```

FMRI Segment	Description
pkg://	FMRI scheme (other: svc, lrc)
solaris	Publisher
developer/gcc	Package name
5.4.0	Component version
5.11	Build version / OS kernel version
0.175.3.22.0.1.0	Branch version. Only meaningful to the application developer
20170620T160441Z	ISO 8601 UTC format (YYYYMMDDTHHMMSS)

Displaying the Contents of a Package

To display information about package contents, run `pkg contents` *`pkg-fmri`*.

```
# pkg contents compress/zip
PATH
usr
usr/bin
usr/bin/zip
usr/bin/zipcloak
usr/bin/zipnote
usr/bin/zipsplit
usr/share
usr/share/man
<output truncated>
```

Updating an Installed Package

To update an installed package, run `pkg update` *`pkg-fmri`*.

```
# pkg update compress/zip
No updates available for this image.
```

Updating a System

- Update a system in a single step.
- Use of pkg update –nv to do a dry run

```
# pkg update --accept
-------------------------------------------------------
Package: pkg://solaris/consolidation/osnet/osnet-incorporation@0.5.11,5.11-
0.175.3.32.0.4.0:20180426T184953Z
License: lic_OTN

You acknowledge that your use of this Oracle Solaris software product
...
Packages to remove:  27
         Packages to install:  34
          Packages to update: 451
          Packages to change:   1
         Mediators to change:   1
      Create boot environment: Yes
Create backup boot environment:  No
...
```

Viewing an Installation Action Without Installing

To view an installation action without installing the package, run `pkg install -n` *`pkg-fmri`*.

```
# pkg install -nv apptrace
            Packages to install:                 1
     Estimated space available:          31.96 GB
Estimated space to be consumed:          19.83 MB
       Create boot environment:                No
Create backup boot environment:                No
          Rebuild boot archive:                No

Changed packages:
solaris
   developer/apptrace
     None -> 0.5.11,5.11-0.175.3.0.0.27.1:20150717T222030
```

Installing a Package

To install a package, run `pkg install` *`pkg-fmri`*.

```
# pkg install apptrace
              Packages to install:     1
          Create boot environment:    No
   Create backup boot environment:    No

DOWNLOAD                PKGS       FILES    XFER (MB)  SPEED
Completed                1/1      10/10     0.1/0.1    31.6k/s

PHASE                                        ITEMS
Install new actions                         29/29
Updating package state database              Done
Updating package cache                        0/0
Updating image state                         Done
Creating fast lookup database                Done
Updating package cache                        1/1
```

Installing a Package

```
# pkg install developer/gcc

# pkg install pkg:/developer/gcc

# pkg install pkg://solaris/developer/gcc

# pkg install pkg://solaris/developer/gcc@5.4.0

# pkg install pkg://solaris/developer/gcc-3@5.4.0,5.11-0.175.3.22.0.1.0:20170620T160441Z
```

Verifying a Package Installation

To verify a package installation, run `pkg verify pkg-fmri`.

```
# pkg verify -v apptrace
PACKAGE                                                 STATUS
pkg://solaris/developer/apptrace                            OK
```

Searching for a Package

To search for a package, run `pkg search` *`pattern`*.

```
# pkg search -l bash
INDEX                   ACTION VALUE               PACKAGE
basename                dir    etc/bash            pkg:/shell/bash@4.1.11-0.175.3.0.0.27.0
basename                dir    usr/share/bash      pkg:/shell/bash@4.1.11-0.175.3.0.0.27.0
basename                file   usr/bin/bash        pkg:/shell/bash@4.1.11-0.175.3.0.0.27.0
com.oracle.info.name    set    bash                pkg:/shell/bash@4.1.11-0.175.3.0.0.27.0
pkg.fmri                set    solaris/shell/bash  pkg:/shell/bash@4.1.11-0.175.3.0.0.27.0
```

Searching for a Package

Use option and wildcard to return different result.

```
# pkg search -l /usr/bin/bash

# pkg search –o pkg.name /lib/libpower.so.1

# pkg search –H */solaris-desktop

# pkg search –l –o pkg.name ?trace

# pkg search –H –o fmri '*/solaris-desktop:depend:group:'
```

Uninstalling a Package

To uninstall a package, run `pkg uninstall` *`pkg-fmri`*.

```
# pkg uninstall apptrace
               Packages to remove:     1
          Create boot environment:    No
    Create backup boot environment:    No

PHASE                                          ACTIONS
Removing old actions                             25/25
Updating package state database                    Done
Updating package cache                              1/1
Updating image state                               Done
Creating fast lookup database                      Done
Updating package cache                              1/1
```

Package Management Commands: Summary

Package Management Task	IPS Command
Display package state and version information	`pkg list`
Display package information	`pkg info`
Display the contents of a package	`pkg contents`
Install package updates	`pkg update`
Install the package	`pkg install`
Verify the package installation	`pkg verify`
Search for a package	`pkg search`
Uninstall a package	`pkg uninstall`

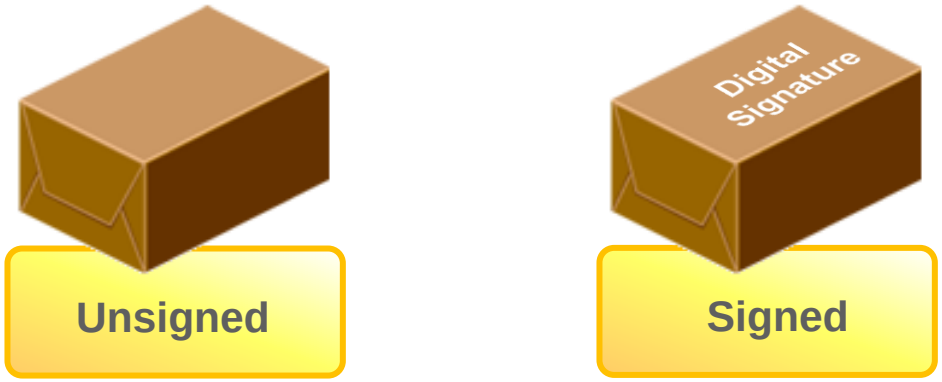
Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- Managing Software Packages
- **Managing Signed Packages and Package Properties**
- Describing the use of IPS in packaging the Oracle Solaris OS
- Managing Boot Environments
- Updating the OS by using IPS

Introducing Signed Packages

Signed packages contain digital signatures that verify that:

- The package came from the entity who signed it
- The entity signed the package
- The package has not been modified
- The entity is trusted



Installing Signed Packages

- Identifying image properties for signed packages
- Configuring image properties for signed packages
- Identifying publisher properties for signed packages
- Configuring publisher properties for signed packages

Identifying Image Properties for Signed Packages

Property	Description
signature-policy	Determines what checks will be performed on manifests when you install a package into the operating system image: • ignore • verify • require-signatures • require-names
signature-required-names	Defines names that must be seen as common names of certificates while validating the signatures of a package
trust-anchor-directory	Identifies the path name of the directory that contains the trust anchors for the image

Configuring Image Properties for Signed Packages

Use `pkg` with the following subcommands to configure package signature properties for an image:

- `set-property`
- `add-property-value`
- `remove-property-value`
- `unset-property`

Examples:

```
# pkg set-property signature-policy verify
# pkg add-property-value signature-require-names trustedname
# pkg remove-property-value signature-require-names trustedname
# pkg unset-property signature-policy
```

Identifying Publisher Properties for Signed Packages

Property	Description
`signature-policy`	Determines the checks that will be performed on manifests when installing a package into the image from a specified publisher
`signature-required-names`	Defines names that must be seen as common names of certificates while validating the signatures of a package from a specified publisher

Configuring Publisher Properties for Signed Packages

Use `pkg set-publisher` with the following subcommands to configure package signature properties for a publisher:

- `--set-property`
- `--add-property-value`
- `--remove-property-value`
- `--unset-property`

Examples:

```
# pkg set-publisher --set-property signature-policy=require-signatures
whoisit.com
# pkg set-publisher --add-property-value signature-require-
names=trustedname whoisit.com
# pkg set-publisher --remove-property-value signature-require-
names=trustedname whoisit.com
# pkg set-publisher --unset-property signature-policy whoisit.com
```

Introducing Variants and Facets

- **Variant:** Mutually exclusive component of a package
  - Appears as a tag on IPS actions
  - Affects whether an IPS action is installable
  - **All** variant tags must match
- **Facet:** Optional component of a package
  - Appears as a tag on IPS actions
  - Affects whether an IPS action is installable
  - **Any** facet tag must match

Displaying and Changing Variants

Variants

- To display the values of all variants, use `pkg variant`.
- To display specific variants:
  - use `pkg variant`
  - or `pkg variant arch`
- To change a variant:
  - use `pkg change-variant -n --accept` *`variant_spec=instance`*.
  - example `pkg change-variant -n --accept` *`arch=sparc`*

. NOTE: -n for dry run

Displaying and Changing Facets

Facets

- To display the current values of all facets defined in the current image, use `pkg facet`.
- To display specific facets:
  - `pkg facet`
  - `pkg facet local.zh`
- To change the current value of a facet:
  - `pkg change-facet -n --accept locale.zh=false`
  - `pkg change-facet -n locale.*=false`

Other facet tags

```
facet.devel                    facet.doc
facet.doc.html                 facet.doc.info
facet.doc.man                  facet.doc.pdf
facet.locale.de                facet.locale.en_GB
facet.locale.en_US             facet.locale.fr
facet.locale.ja_JP             facet.locale.zh_CN
```

Managing Package History

- To view package history, run `pkg history`.
- To view verbose package history information, run `pkg history -l`.
- To specify the number of the most recent package history operations to display, use the `-n` option.
- To display log records for a comma-separated list of time stamps, use the `-t` option.
- To purge package history, run `pkg purge-history`.

```
$ pkg history
START                 OPERATION            CLIENT               OUTCOME
2015-06-24T15:20:16   set-property     transfer module      Succeeded
2015-06-24T15:20:16   images-create    transfer module      Succeeded
2015-06-24T15:20:16   add-publisher    transfer module      Succeeded
…
# pkg purge-history
```

Package History : specifying time

- use history -t <time format>
- time format must be in %Y-%m-%dT%H:%M:%S
- List pkg operation at exact date and time

```
$ pkg history -t 2018-06-24T10:45:05 -l
...
```

- List all pkg operations between 9am on 24-Jun and 10am on 26-Jun

```
$ pkg history -t 2018-06-24T09:00:00-2018-06-26T10:00:00
START                 OPERATION         CLIENT                OUTCOME
2015-06-24T15:20:16   set-property      transfer module       Succeeded
2015-06-24T15:20:16   images-create     transfer module       Succeeded
2015-06-24T15:20:16   add-publisher     transfer module       Succeeded
…
```

Quiz

Which command enables you to configure your current image to ensure that all manifests with signatures are validly signed?

a. `# pkg set-property signature-policy verify`
b. `# pkg set-property signature-policy require-names`
c. `# pkg set-property signature-policy require-signature`

Quiz

Which command enables you to configure your current image to ensure that all manifests with signatures are validly signed?

a. `# pkg set-property signature-policy verify`

b. `# pkg set-property signature-policy require-names`

c. `# pkg set-property signature-policy require-signature`

Quiz

Which `pkg subcommand` and `parameter` is used to configure publisher properties for signed packages? [Choose two]

a. `set-property`
b. `set-publisher`
c. `set-publisher property`
d. `--set-publisher`
e. `--set-property`

Quiz

Which `pkg subcommand` and `parameter` is used to configure publisher properties for signed packages? [Choose two]

a. `set-property`
b. `set-publisher`
c. `set-publisher property`
d. `--set-publisher`
e. `--set-property`

Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- Managing Software Packages
- Managing Signed Packages and Package Properties
- **Describing the use of IPS in packaging the Oracle Solaris OS**
- Managing Boot Environments
- Updating the OS by using IPS

Use of IPS in Packaging the Oracle Solaris OS

- Oracle uses the following key IPS components to package the Oracle Solaris OS, along with the various package dependencies:
  - FMRI
  - Incorporation packages
  - Group packages

Oracle Solaris Package Versioning: FMRI

FMRI of a sample package:

```
pkg://solaris/system/library/storage/suri@0.5.11,5.11-
0.175.3.0.0.30.0:20150821T155633Z
```

- `0.5.11`: Specifies the OS `major.minor` version
- `5.11`: Specifies the OS release that this package was built for
- `0.175.3.0.0.30.0`: Specifies the branch version

Version	Description
`0.175`	Specifies the major OS release number. In this example, `0.175` indicates Oracle Solaris 11.
`3`	Specifies the update release number. In this example, 3 indicates Oracle Solaris 11.3.
`0`	Specifies the SRU number for this update release

Oracle Solaris Package Versioning: FMRI

FMRI of a sample package (continued):

```
pkg://solaris/system/library/storage/suri@0.5.11,5.11-
0.175.3.0.0.30.0:20150821T155633Z
```

Version	Description
0	Refers to a reserved field, which is currently not used for Oracle Solaris packages
30	Specifies the release or SRU build number, or respin number for the major OS release
0	Specifies the build number for the individual nightly builds

- `20150821T155633Z`: Specifies the time stamp when the package is published, in `ISO-8601` basic format: *`YYYMMDD`*`T`*`HHMMSS`*`Z`.

Oracle Solaris Incorporation Packages

- An incorporation package:
  - Specifies the versions of other packages that can be installed
  - Ensures that if you install an `incorporate` dependency package, only the prescribed version of the dependent package can be installed
- Oracle Solaris is delivered by a set of packages, with each group of packages constrained by an incorporation.
- Each incorporation roughly represents the organization that developed each group of packages.

```
$ pkg list *incorporation
```

- The special `entire` incorporation of Oracle Solaris constrains all other incorporations to the same build, such that all packages are upgraded as a single group.

Oracle Solaris Group Packages

- A group package specifies the set of packages that constitute a feature or tool.
- Installing a group package installs all the `group` dependency packages in that group package.
- Oracle Solaris defines several group packages that contain `group` dependencies.
- These group packages enable convenient installation of common sets of packages.

```
$ pkg list -Has 'pkg:/group/*'
```

- For example, the `group/system/solaris-minimal-server` package delivers the set of packages required for the minimum supported Oracle Solaris environment.

Avoiding Installing Some Packages in a Group Package

Use the `pkg avoid` command to avoid installing specified packages if they are the target of a group dependency.

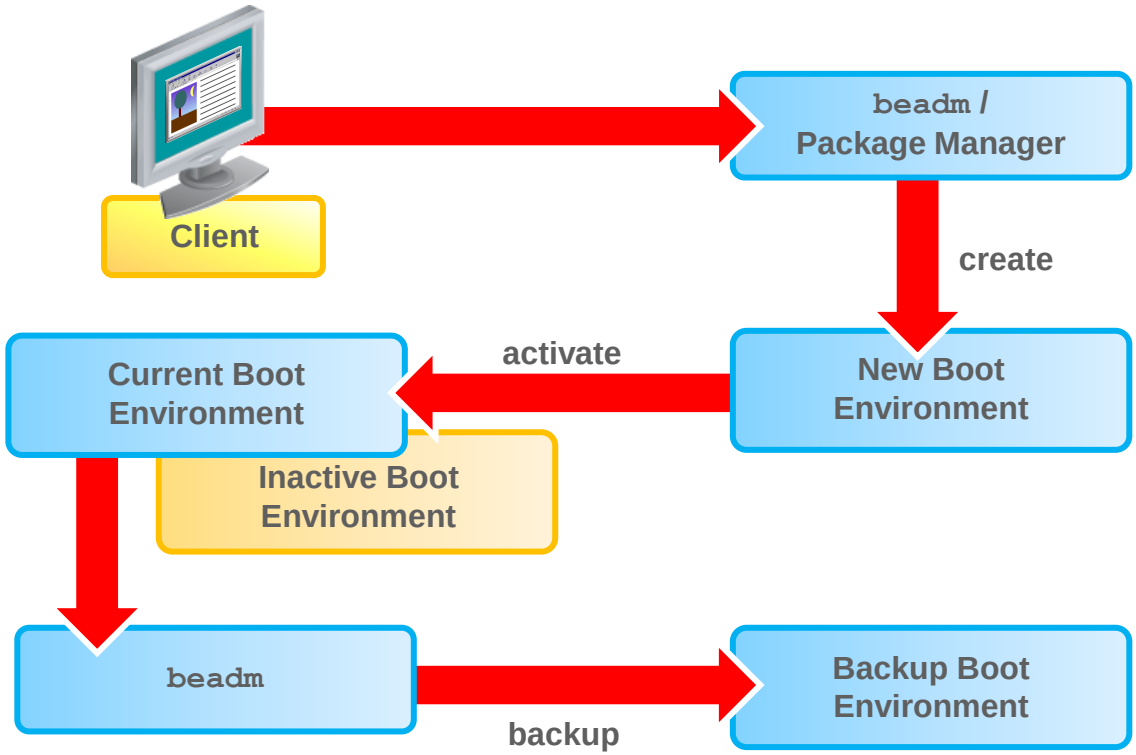
```
$ pkg list -a group/feature/amp
NAME (PUBLISHER)                                VERSION                        IFO
group/feature/amp                               0.5.11-0.175.2.0.0.33.0        ---
$ pkg list -a `pkg contents -o fmri -Hrt depend -a type=group group/feature/amp`
NAME (PUBLISHER)                                VERSION                        IFO
database/mysql-51                               5.1.37-0.175.2.0.0.34.0        ---
web/php-52                                      5.2.17-0.175.2.0.0.34.0        i--
web/php-52/extension/php-apc                    3.0.19-0.175.2.0.0.34.0        i--
web/php-52/extension/php-mysql                  5.2.17-0.175.2.0.0.34.0        i--
web/server/apache-22                            2.2.26-0.175.2.0.0.34.0        i--
web/server/apache-22/module/apache-dtrace       0.3.1-0.175.2.0.0.34.0         ---
web/server/apache-22/module/apache-fcgid        2.3.9-0.175.2.0.0.34.0         ---
web/server/apache-22/module/apache-php5         5.2.17-0.175.1.0.0.18          --r
```

```
$ pkg avoid apache-fcgid
$ pkg avoid
web/server/apache-22/module/apache-fcgid
```

Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- Managing Software Packages
- Managing Signed Packages and Package Properties
- Describing the use of IPS in packaging the Oracle Solaris OS
- **Managing Boot Environments**
- Updating the OS by using IPS

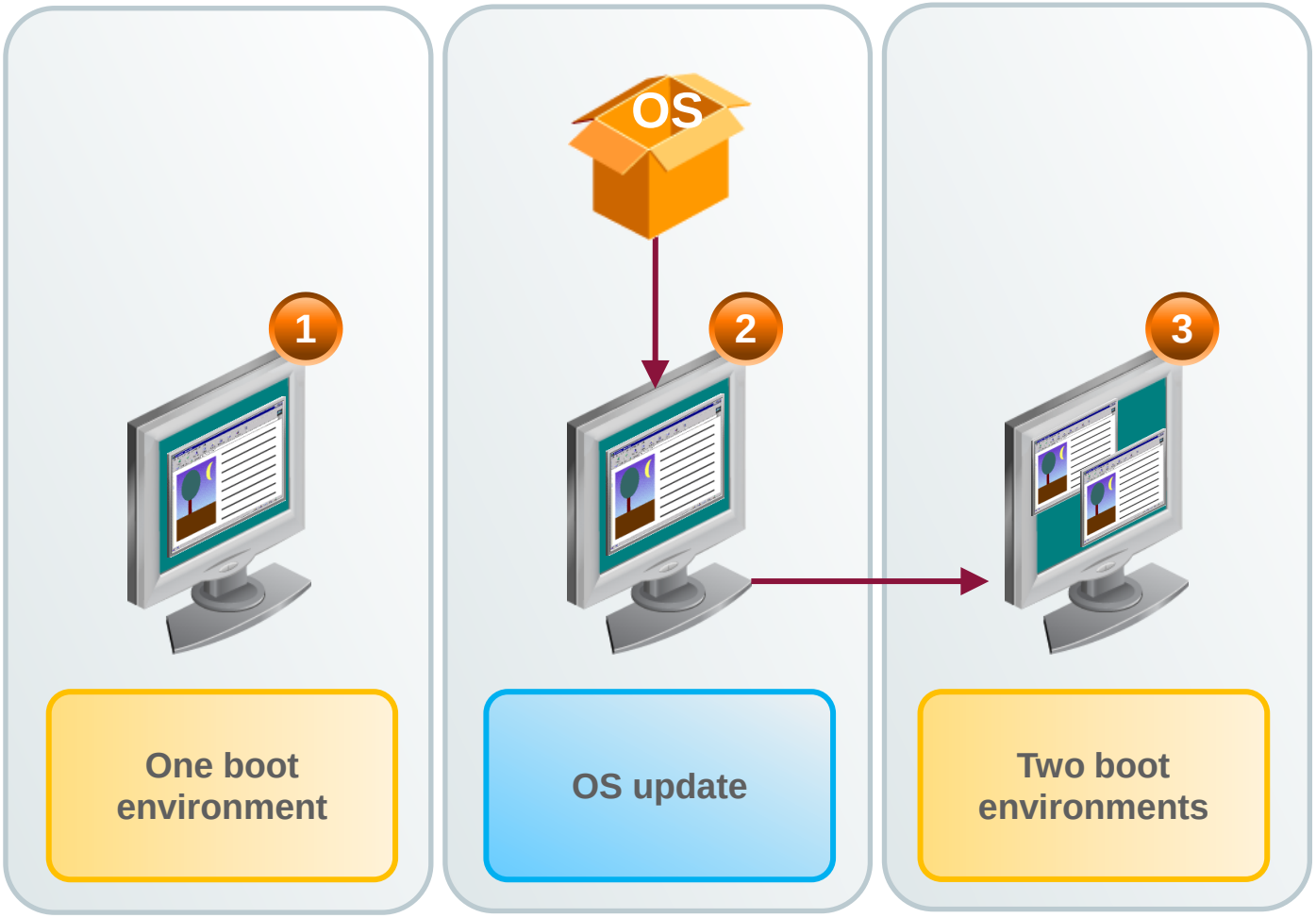
Managing Boot Environments



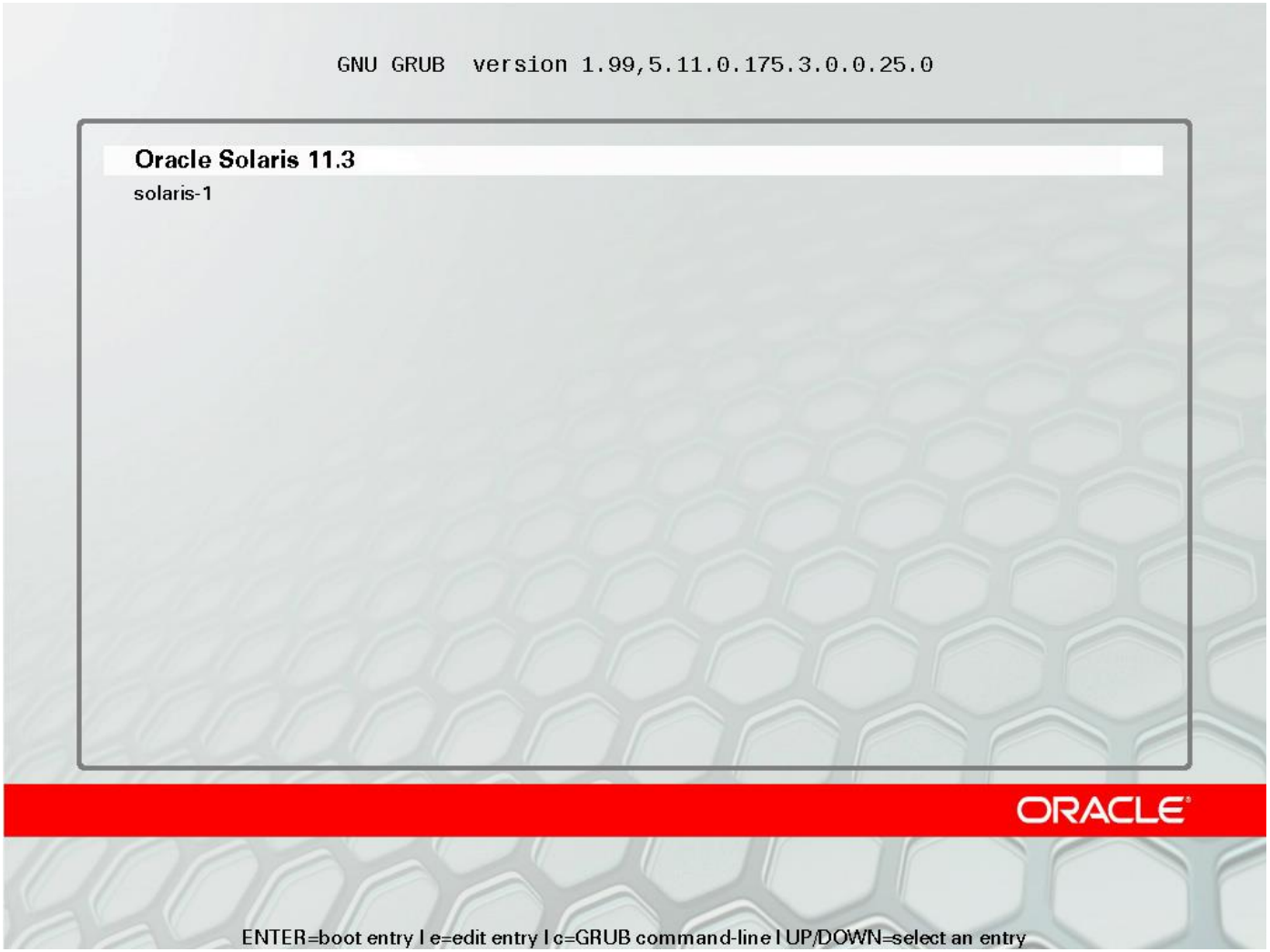
Boot environment management utilities:

- `beadm` command
- Package Manager

Creating a Boot Environment: Example



Creating a Boot Environment: GRUB menu



Managing Boot Environments with the `beadm` Command

In this section, you learn how to perform the following tasks by using the `beadm` command:

- Listing a new boot environment
- Creating a new boot environment
- Renaming an existing inactive boot environment
- Destroying an existing inactive boot environment
- Activating an existing inactive boot environment
- Verifying the new boot environment
- Mounting an inactive boot environment
- Installing a package on an inactive mounted boot environment
- Uninstalling a package on an inactive mounted boot environment
- Unmounting an inactive boot environment
- Creating a backup of a boot environment
- Creating a boot environment from an existing backup

Listing the Boot Environments on the System

Run `beadm list`.

```
# beadm list
BE        Active Mountpoint Space    Policy Created
--        ------ ---------- -----    ------ -------
solaris   NR     /          5.35G    static 2015-02-12 11:11
solaris-1 -      -          140.45M  static 2015-02-18 08:02
solaris-2 -      -          176.0K   static 2015-02-18 10:31
```

Creating a New Boot Environment

Run `beadm create` *beName*.

```
# beadm create test1
# beadm list
BE        Active Mountpoint Space    Policy Created
--        ------ ---------- -----    ------ -------
solaris   NR     /          5.35G    static 2015-02-12 11:11
solaris-1 -      -          140.45M  static 2015-02-18 08:02
solaris-2 -      -          176.0K   static 2015-02-18 10:31
test1     -      -          67.0K    static 2015-03-01 01:34
#
```

Renaming an Existing Inactive Boot Environment

Run `beadm rename` *beName newBeName*.

```
# beadm rename test1 apptest1
# beadm list
BE        Active Mountpoint Space    Policy Created
--        ------ ---------- -----    ------ -------
apptest1  -      -          67.0K    static 2015-03-01 01:34
solaris   NR     /          5.35G    static 2015-02-12 11:11
solaris-1 -      -          140.45M  static 2015-02-18 08:02
solaris-2 -      -          176.0K   static 2015-02-18 10:31
#
```

Destroying an Existing Inactive Boot Environment

Run `beadm destroy` *beName*.

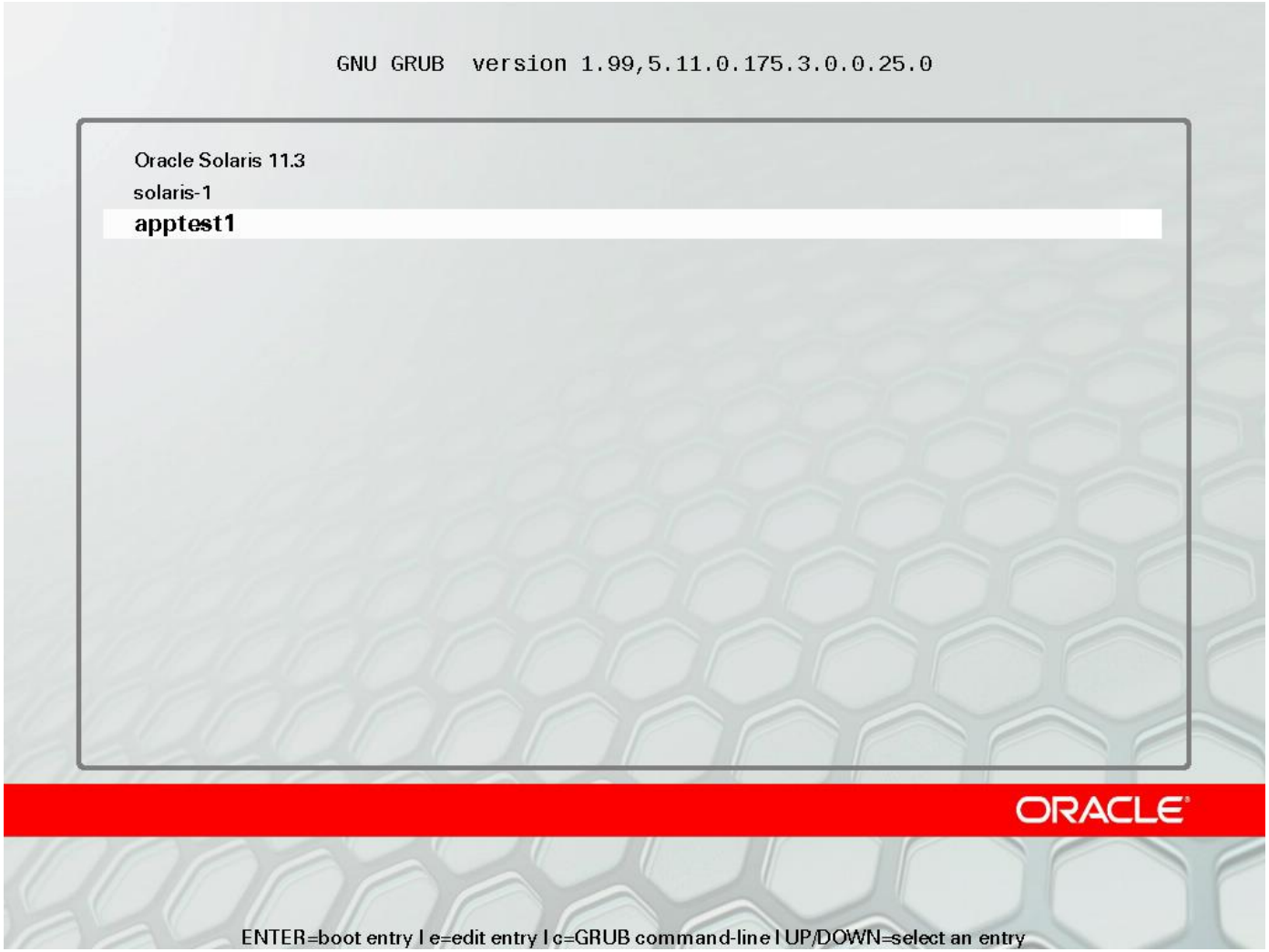
```
# beadm destroy solaris-2
Are you sure you want to destroy solaris-2? This action cannot
be undone (y/[n]): y
# beadm list
BE        Active Mountpoint Space    Policy Created
--        ------ ---------- -----    ------ -------
apptest1  -      -          67.0K    static 2015-03-01 01:34
solaris   NR     /          5.35G    static 2015-02-12 11:11
solaris-1 -      -          140.45M  static 2015-02-18 08:02
#
```

Activating an Existing Inactive Boot Environment

Run `beadm activate` *beName*.

```
# beadm activate apptest1
# beadm list
BE        Active Mountpoint Space    Policy Created
--        ------ ---------- -----    ------ -------
apptest1  R      -          67.0K    static 2015-03-01 01:34
solaris   N      /          5.35G    static 2015-02-12 11:11
solaris-1 -      -          140.45M  static 2015-02-18 08:02
# init 6
```

Verifying the New Boot Environment



Mounting an Inactive Boot Environment

Run `beadm mount` *`beName mountpoint`*.

```
# beadm mount solaris-1 /solaris-1
# beadm list
BE        Active  Mountpoint Space  Policy  Created
--        ------  ---------- -----  ------  -------
solaris   NR      /            2.38G   static 2015-03-08 03:50
solaris-1 -       /solaris-1 169.01M   static 2015-03-10 22:14
```

Installing a Package on an Inactive Mounted Boot Environment

Run `pkg -R` *`mountpoint`* `install` *`packagename`*.

```
# pkg -R /solaris-1 install diffstat
Creating plan...
              Packages to install:    1
           Create boot environment:    No
    Create backup boot environment:    No
DOWNLOAD                    PKGS       FILES    XFER (MB) SPEED
Completed                    1/1         6/6     0.0/0.0  21.4k/s

PHASE                                ITEMS
Installing new actions               24/24
Updating package state database       Done
Updating package cache                0/0
Updating image state                  Done
Creating fast lookup database         Done
```

Uninstalling a Package on an Inactive Mounted Boot Environment

Run `pkg -R` *`mountpoint`* `uninstall` *`packagename`*.

```
# pkg -R /solaris-1 uninstall diffstat
```

Unmounting an Inactive Boot Environment

Run `beadm unmount` *beName*.

```
# beadm unmount solaris-1
# beadm list
BE        Active  Mountpoint Space  Policy  Created
--        ------  ---------- -----  ------  -------
solaris   NR      /            2.38G   static 2015-03-08 03:50
solaris-1 -       -          170.01M   static 2015-03-10 22:14
```

Creating a Backup of a Boot Environment

Run `beadm create` *`BEname@snapshotdescription`*.

```
# beadm create solaris@backup
# beadm list -a solaris
BE/Dataset/Snapshot                          Active Mountpoint Space   Policy Created
-------------------                          ------ ---------- -----   ------ -------
solaris
   rpool/ROOT/solaris                        NR     /          2.27G   static 2015-03-08 11:32
   rpool/ROOT/solaris/var                    -      /var       112.37M static 2015-03-08 11:32
   rpool/ROOT/solaris/var@2015-03-08-12:17:23 -     -          760.5K  static 2015-03-08 17:47
   rpool/ROOT/solaris/var@backup             -      -          24.0K   static 2015-03-08 18:06
   rpool/ROOT/solaris/var@install            -      -          18.86M  static 2015-03-08 11:37
   rpool/ROOT/solaris@2015-03-08-12:17:23    -      -          50.27M  static 2015-03-08 17:47
   rpool/ROOT/solaris@backup                 -      -          0       static 2015-03-08 18:06
   rpool/ROOT/solaris@install                -      -          53.28M  static 2015-03-08 11:37
```

Creating a Boot Environment from an Existing Backup

Run `beadm create -e` *`BEname@snapshotdescription beName`*.

```
# beadm create -e solaris@backup solaris-2
# beadm list
BE        Active Mountpoint Space  Policy Created
--        ------ ---------- -----  ------ -------
solaris   NR     /          2.38G  static 2015-03-08 03:50
solaris-1 -      -        170.01M  static 2015-03-10 22:14
solaris-2 -      -          28.0K  static 2015-03-10 22:59
# beadm activate solaris-2
# beadm list
BE        Active Mountpoint Space  Policy Created
--        ------ ---------- -----  ------ -------
solaris   N     /           2.38G  static 2015-03-08 03:50
solaris-1 -      -        170.01M  static 2015-03-10 22:14
solaris-2 R      -          28.0K  static 2015-03-10 22:59
# init 6
```

Quiz

Which utility is used to manage boot environments in Oracle Solaris 11?

a. Luupgrade

b. Beadm

c. BE Manager

Quiz

Which utility is used to manage boot environments in Oracle Solaris 11?

a. Luupgrade

b. Beadm

c. BE Manager

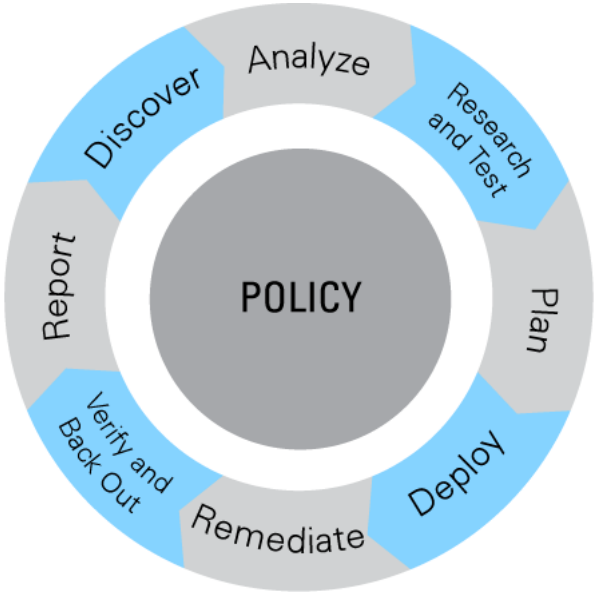
Agenda

- Describing IPS, Its Components, and Interfaces
- Configuring an IPS Client to Access the Local IPS Repository
- Managing Package Publishers
- Managing Software Packages
- Managing Signed Packages and Package Properties
- Describing the use of IPS in packaging the Oracle Solaris OS
- Managing Boot Environments
- **Updating the OS by using IPS**

Planning for an Oracle Solaris 11 OS Software Update

Planning is required to make sure that the operating system is:

- Running the most current version of the software
- Updated regularly to make sure that the latest feature updates and bug fixes have been installed
- Updated in accordance with business policy, software update strategy, and plan

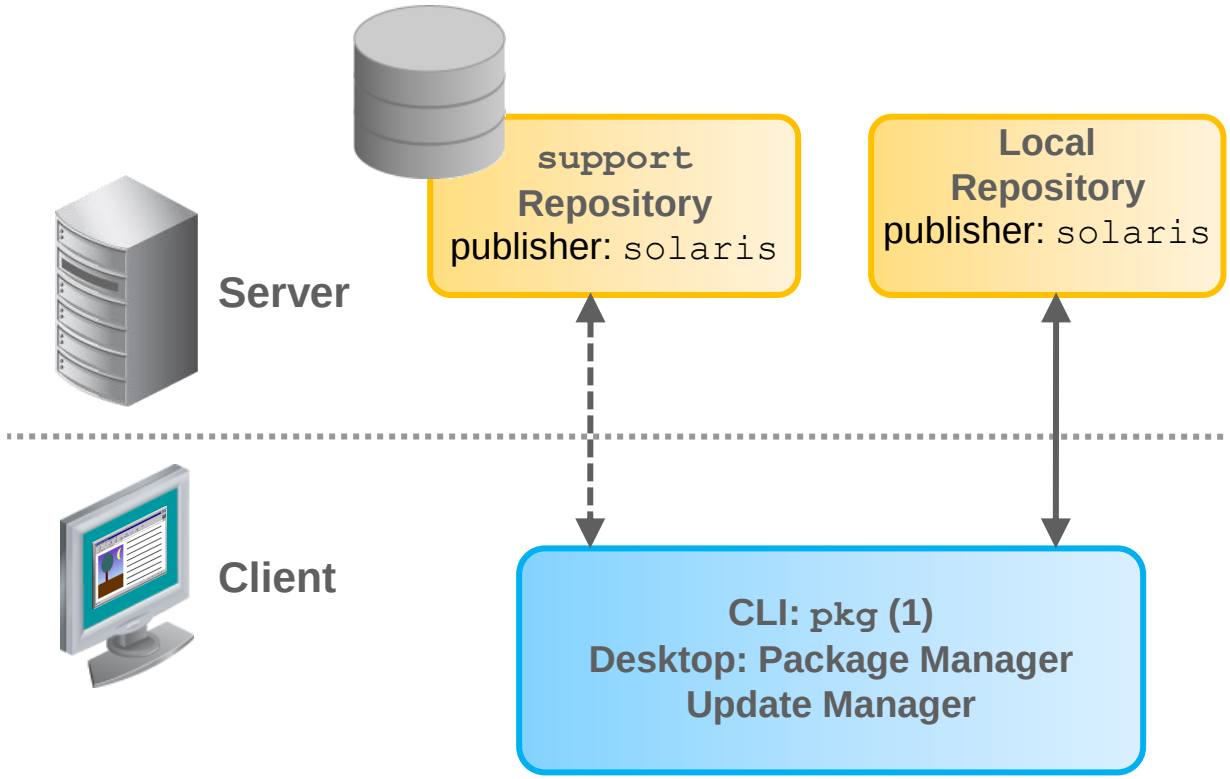


Software Update Plan

The software update plan should provide you with the following information:

- Who will perform the update
- When the update should be performed
- What systems should be updated
- What software packages should be updated
- How the update should be performed

Software Update Process: Overview



Support repository: http://pkg.oracle.com/solaris/support/
Local repository: Configured on your local network

Updating a System to latest Oracle Solaris

You can update your system to the latest Oracle Solaris OS by using one of the following repositories:

- Web-based Oracle Solaris support repository, for those who have an Oracle support agreement
  https://pkg.oracle.com/solaris/support
- Web-based Oracle Solaris release repository, for those without an Oracle support agreement
  https://pkg.oracle.com/solaris/release

Determining Your Starting Point

Before you start updating your system, determine your starting point.

- Identify the repository that is in use on your system.
- Identify the OS that your system is running.
- Verify the SRU that your system is running.

Identifying the Repository in Use

For a system without Oracle Support Agreement, and using the Oracle Solaris release repository, you will see the following output:

```
# pkg publisher
PUBLISHER                TYPE    STATUS P LOCATION
solaris                  origin  online F http://pkg.oracle.com/solaris/release
```

For a system with Oracle Support Agreement, and using the Oracle Solaris support repository, you will see the following output:

```
# pkg publisher
PUBLISHER                TYPE    STATUS P LOCATION
solaris                  origin  online F http://pkg.oracle.com/solaris/support
```

Identifying the OS Version Currently Installed

Use the `uname -a` command to verify the OS that is currently installed on your system.

```
# uname -a
SunOS t4-1ou 5.11 11.1 sun4v sparc sun4v
```

In this example, Oracle Solaris 11.1 is the installed OS.

Verifying the SRU Currently Installed

If your system has Oracle Solaris 11 11/11 installed with an SRU, the package information appears as follows:

```
% pkg info entire
       Name: entire
       Summary: entire incorporation including Support Repository Update
   Description: This package constrains system package versions to the same build.
                WARNING: Proper system update and correct package selection depend on the
presence of this incorporation. Removing this package will result in an unsupported system.
For more information see
https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=REFERENCE&id=1372094.1.
      Category: Meta Packages/Incorporations
         State: Installed
     Publisher: solaris
       Version: 0.5.11
 Build Release: 5.11
        Branch: 0.175.0.10.0.5.0
Packaging Date: Fri Aug 03 18:26:27 2012
          Size: 5.45 kB
          FMRI: pkg://solaris/entire@0.5.11,5.11-0.175.0.10.0.5.0:20120803T182627Z
```

In this example, SRU #10.5 is installed.

Updating a System Running Oracle Solaris 11 11/11 to Oracle Solaris 11.3 by Using the Release Repository

1. Ensure that your package repository is set to use https://pkg.oracle.com/solaris/release.
   # pkg publisher
2. Review the licenses of the Oracle Solaris 11.1 pre-upgrade repository (`0.5.11-0.175.0.10.1.0.0`).
   # pkg update --license
3. Update the system packages. A new boot environment is created.
   # pkg update –be-name Sol11Upg --accept entire@0.5.11-0.175.0.10.1.0.0
4. Reboot the system to use the new updated boot environment.
   # init 6
5. On a SPARC system only, for each zone, remove the `ldomsmanager` package.
   # for z in `zoneadm list`; do zlogin $z pkg uninstall ldomsmanager; done

6. Update the IPS package.
   # pkg update package/pkg
7. Review the licenses of the Oracle Solaris 11.3 OS.
   # pkg update --license|less
8. Update the system packages. A new boot environment is created.
   # pkg update –-be-name Solaris11.3 –-accept entire@0.5.11-0.175.3.1.0.15.0
9. Reboot the system to use the new updated boot environment.
   # init 6

Updating a System Running Oracle Solaris 11 11/11 with an SRU to Oracle Solaris 11.3

1. Ensure that your package repository is set to use https://pkg.oracle.com/solaris/support.

   If your system is installed with an SRU earlier than SRU #10.5, update the system to SRU #10.5 (`0.5.11,5.11-0.175.0.10`) and reboot.

   If your system is already installed with an SRU #10.5 or later, continue updating the system to Oracle Solaris 11.3.

   ```
   # pkg update -be-name S11SRU10.5 --accept entire@0.5.11,5.11-0.175.0.10
   # reboot
   ```

2. On a SPARC system only, for each zone, remove the `ldomsmanager` package.
3. Update the IPS packages.
4. Review the licenses of the Oracle Solaris 11.3 update.
5. Update the system to the Oracle Solaris 11.3 OS.
6. Reboot the system to use the new updated boot environment.

Updating a System Running Oracle Solaris 11.1 or 11.2 to the Oracle Solaris 11.3 OS

If Oracle Solaris 11.1 or 11.2 is installed, with or without SRUs, no special steps are required to update the system.

- Become an administrator.
- Update the system packages.
  - If you are using the Oracle Solaris support repository, update the system packages.
  - If you are using the Oracle Solaris release repository, review the update's license.
- Reboot the system to use the new updated boot environment.

Summary

In this lesson, you should have learned how to:

- Describe IPS, its components, and interfaces
- Configure an IPS client to access the local IPS repository
- Manage package publishers
- Manage software packages
- Manage signed packages and package properties
- Describe the use of IPS in packaging the Oracle Solaris OS
- Manage boot environments
- Update the OS by using IPS

Practice 5: Overview

- 5-1: Configuring a Network Client to Access the Local IPS Server
- 5-2: Managing Software Packages by Using the Command-Line Interface
- 5-3: Managing Boot Environments
- 5-4: Updating the OS by Using IPS