Managing Policy-Managed Oracle RAC Databases – Part II

By Ahmed Baraka

Objectives

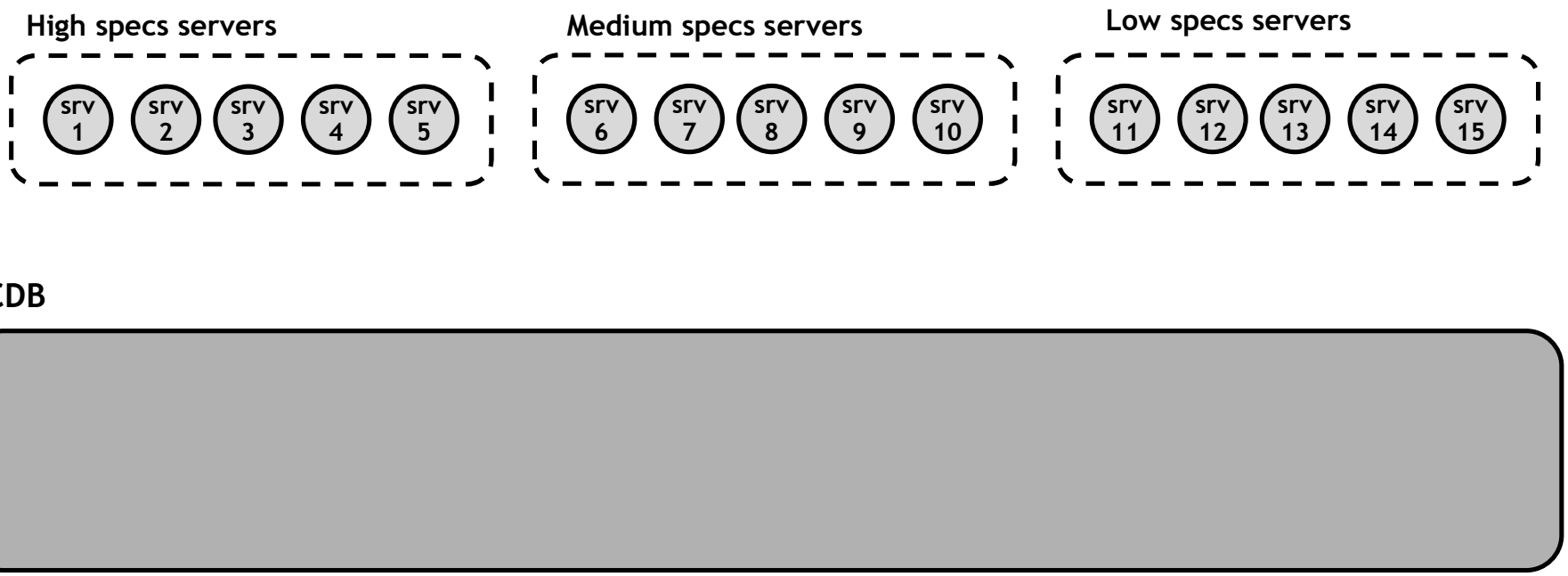
In this lecture, you will learn how to perform the following:

- Describe and create the following components in the Clusterware:
  - Server categorization
  - Cluster configuration policy
  - Cluster configuration policy set
- Obtain and set Server Configuration Attributes

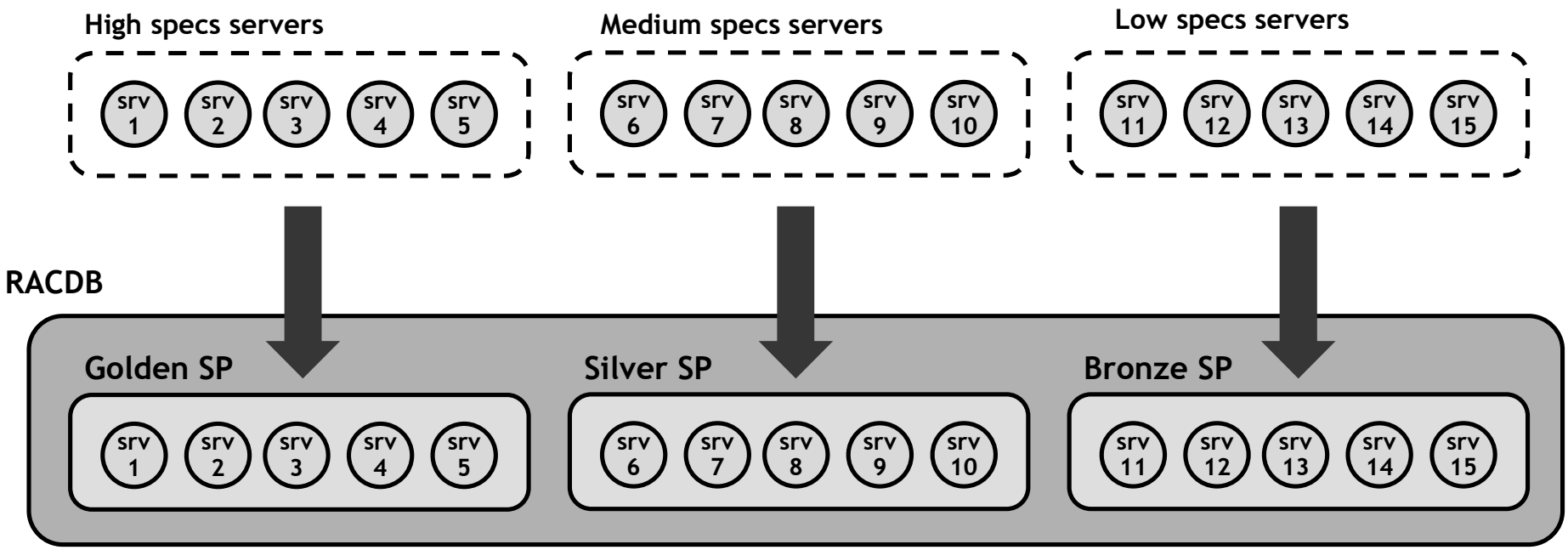
Policy-Management Benefits

- Ensuring database service start order
- Better service failover:
  - Automatically uses any server in the FREE server pool for failover
  - Provides priority to important services for failover

Policy-managed Oracle RAC: Server Categorization

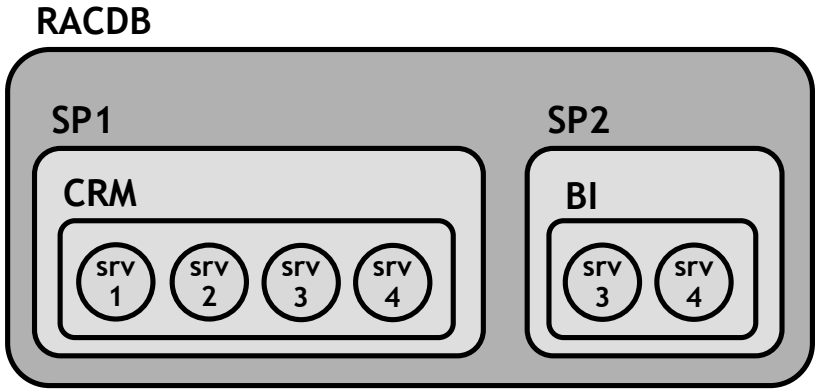


Policy-managed Oracle RAC: Server Categorization



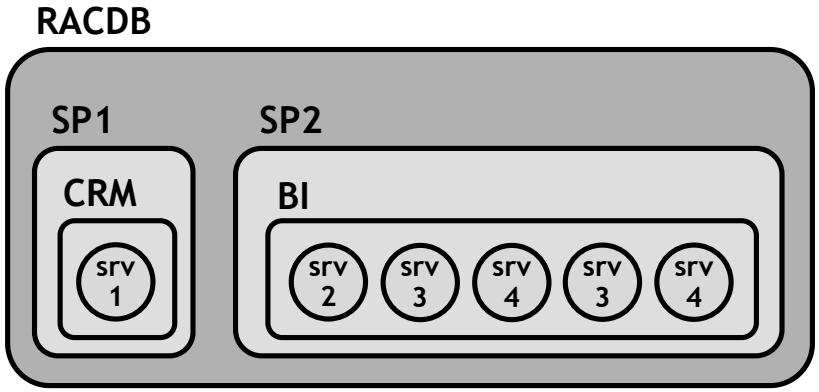
Policy-managed RAC: Dynamic Resource Provisioning

Daytime



Policy-managed RAC: Dynamic Resource Provisioning

Night time

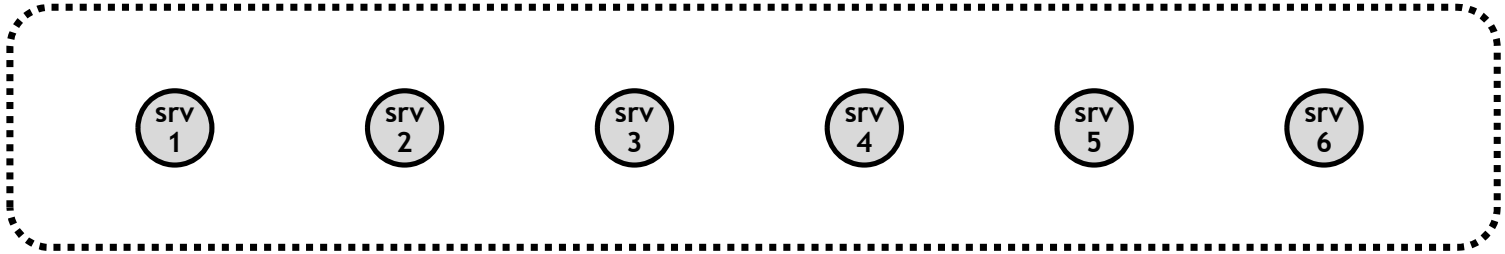


About Server Categorization, Cluster Configuration Policies and the Policy Set

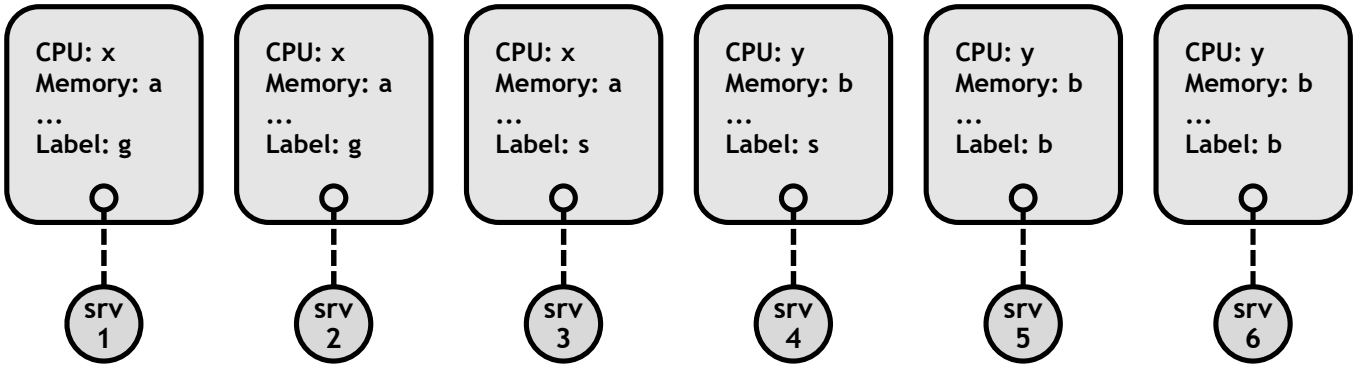
- Server categorization enables you to organize servers into particular categories by using attributes
- Cluster configuration policy is a document that contains exactly one definition for each server pool managed by the cluster configuration policy set
- Cluster configuration policy set is a document which contains one or more configuration policies
- Only one policy is active at a time
- Administrators can set the active policy

Server Categorization

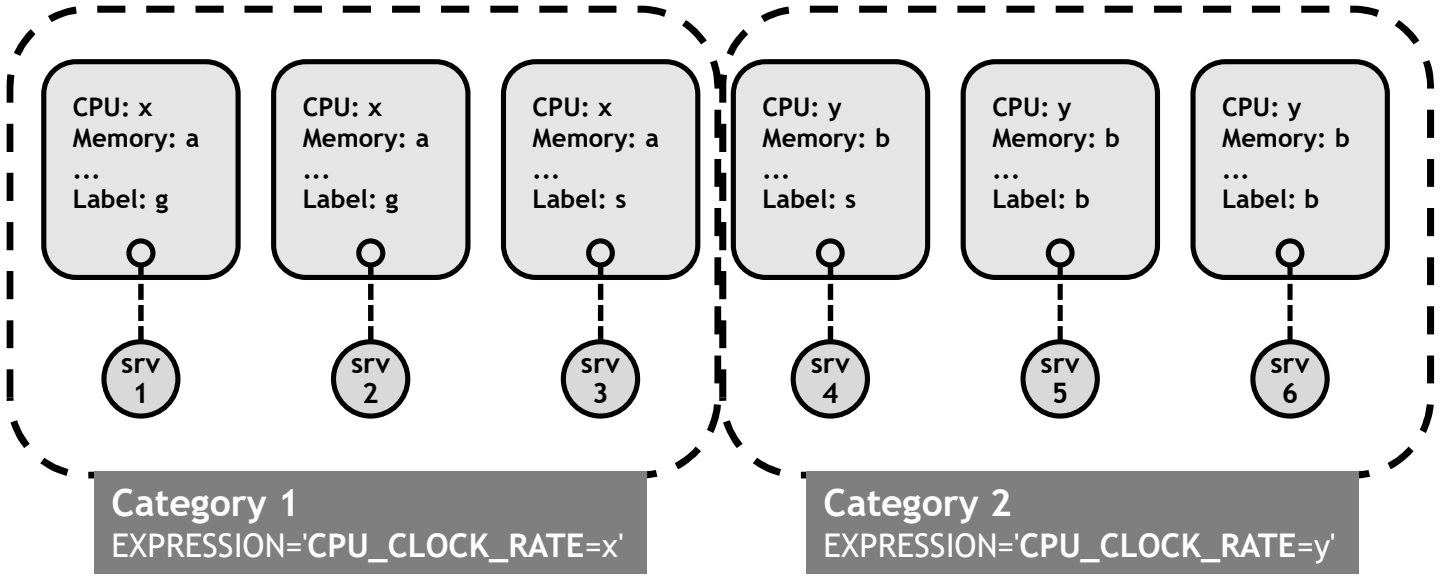
Cluster servers



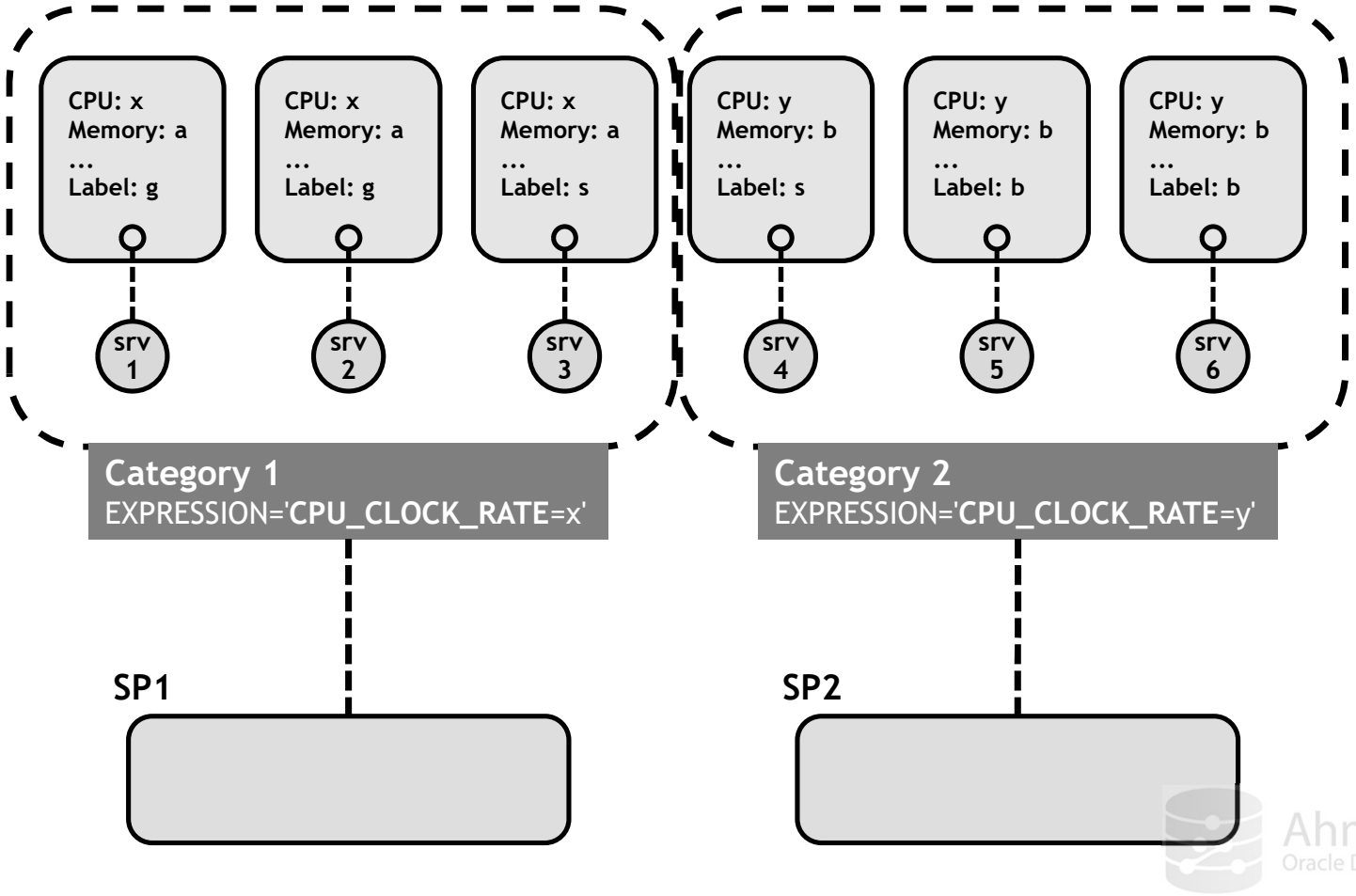
Server Categorization



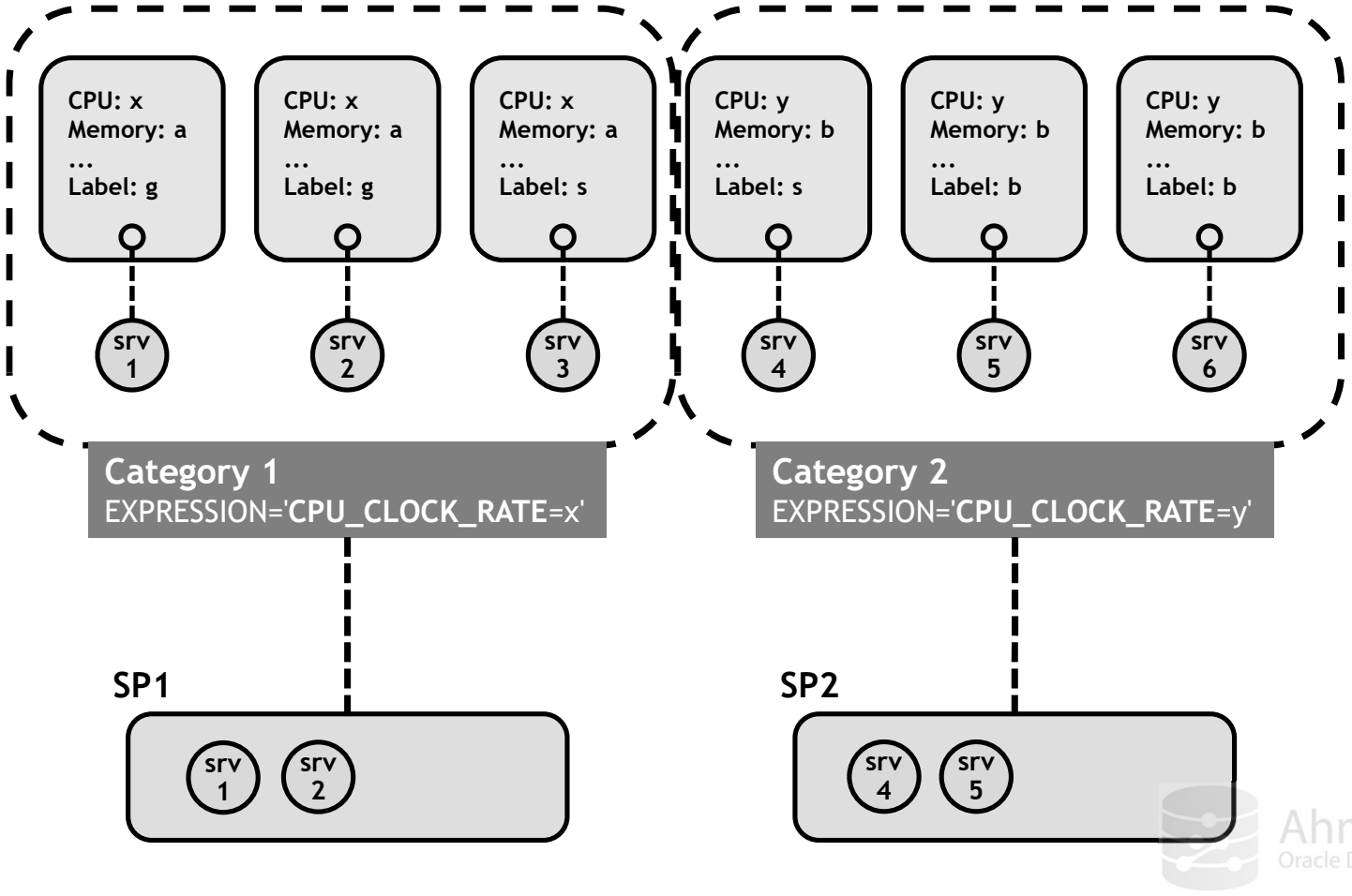
Server Categorization



Server Categorization

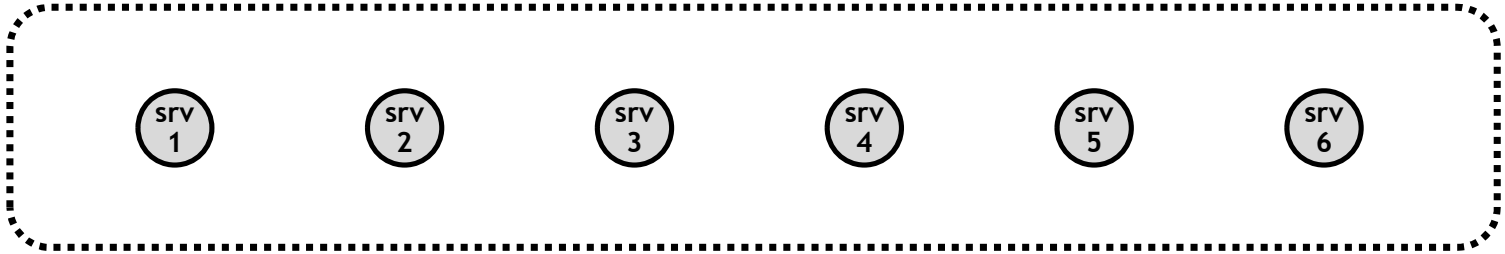


Server Categorization



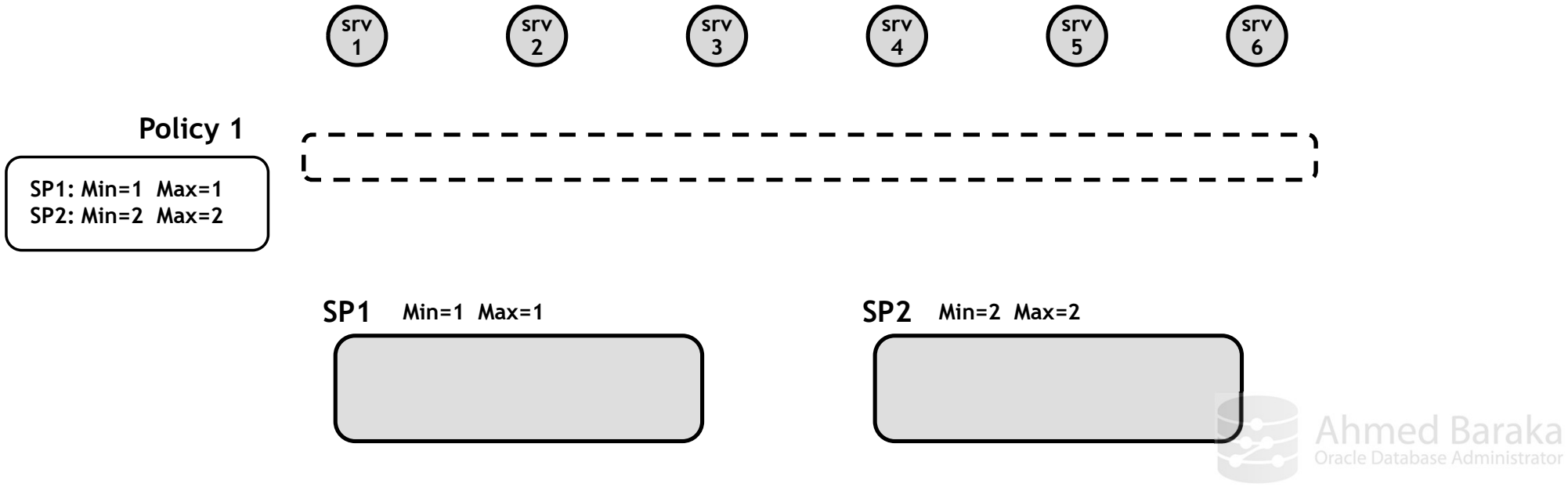
Cluster Configuration Policy

Cluster servers

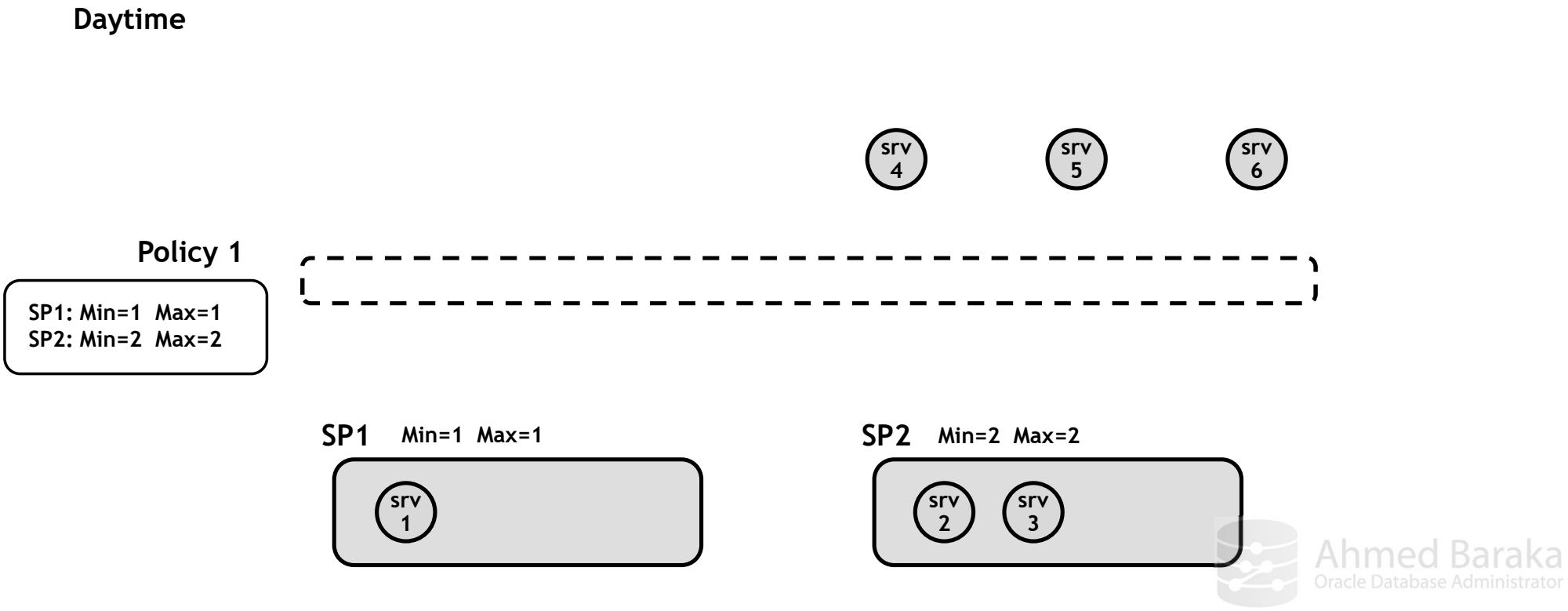


Cluster Configuration Policy

Daytime

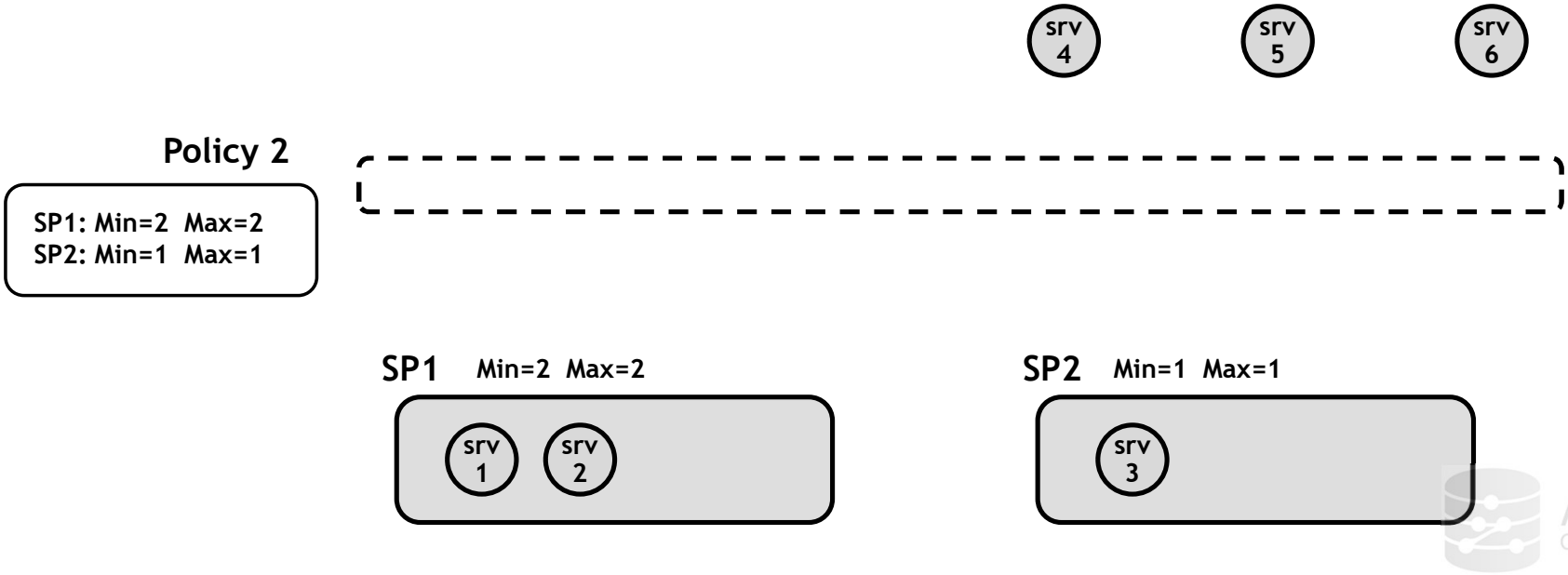


Cluster Configuration Policy



Cluster Configuration Policy

Night time



Policy-Management Benefits

- Ensuring database service start order
- Better service failover:
  - Automatically uses any server in the FREE server pool for failover
  - Provides priority to important services for failover
- Relocate servers based on their categorizations
- Dynamic resource provisioning: relocate the servers based on time or events

Server Configuration Attributes

Attribute	Description
ACTIVE_CSS_ROLE	Role being performed by the server: LEAF or HUB
CONFIGURED_CSS_ROLE	Configured role
CPU_CLOCK_RATE	CPU clock rate in megahertz (MHz)
CPU_COUNT	Number of processors
CPU_EQUIVALENCY	Relative value that describes the CPU power
MEMORY_SIZE	Memory size in megabytes (MB)
RESOURCE_USE_ENABLED	1:server can be moved, 0:server should stay in FREE
SERVER_LABEL	A label that can be set by the user

Setting SERVER_LABEL of a Server

- To set the configuration value of **SERVER_LABEL** to a server, run the following command locally:

```
crsctl set server label GoldS
```

Note: You must restart the Oracle Clusterware technology stack on the node for the changes to take effect.

- To obtain the configuration value of **SERVER_LABEL** server configuration of a server:

```
crsctl get server label
```

Server Category Attributes

Attribute	Description
Name	Name of the server category
EXPRESSION	An expression to determine if a server belongs to the category Acceptable comparison operators include: (=) equal, (eqi): equal case insensitive, (>) greater than, (<) less than, (!=) not equal, (co) contains, (coi) contains case insensitive, (st) starts with, (en) ends with, (nc) does not contain, (nci) does not contain, case insensitive Acceptable boolean operators include: AND, OR `EXPRESSION='((NAME = srv1) OR (NAME = srv2))'`

Create Server Category Examples

- Examples of creating server category:

```
crsctl add category silvercat -attr "EXPRESSION='(CPU_COUNT > 2)
AND (MEMORY_SIZE >2048)'"

crsctl add category highIO
 -attr "EXPRESSION='SERVER_LABEL co IOGold'"
```

- To modify an attribute in a category:

```
crsctl modify category silvercat -attr "EXPRESSION=...
```

- To obtain information about a server category:

```
crsctl status category silvercat
```

Associating a Server Pool to a Category

- Setting the category of a serverpool:

```
srvctl add srvpool gold pool -category goldcat'“
srvctl modify srvpool gold pool -category goldcat'"
```

- To display the associated category of a server pool:

```
crsctl status serverpool goldpool -f | egrep 'CATEGORY'
```

Creating a Policy Set Configuration

Method 1

1. Plan all the required policies
2. Create the new policies:

```
crsctl add policy daytime -attr "DESCRIPTION='Day Time Policy'"
```

3. Set **SERVER_POOL_NAMES** policy set attribute to define the scope of the server pools that are controlled by the policy set

```
crsctl modify policyset -attr "SERVER_POOL_NAME='Free prodpool
devpool testpool'" -ksp
```

Creating a Policy Set Configuration (cont)

Method 1 (cont)

4. Set the attributes for the server pools in every policy

```
crsctl modify serverpool prodpool -attr
"MAX_SIZE=2,MIN_SIZE=2,SERVER_CATEGORY=GoldS" -policy daytime
```

5. Activate the required policy

```
crsctl modify policyset -attr "LAST_ACTIVATED_POLICY='daytime'"
```

Creating a Policy Set Configuration (cont)

Method 2

1. Plan all the required policies
2. Create every policy set in a separate text file
   - The policy defines the server pools it controls and their attributes
3. Register the policies in a policy set
4. Activate the required policy

Creating a Policy Set Configuration: Example

1. Plan your configuration

```
Day Time:
 app1 uses two servers
 app2 use one server

Night Time:
 app1 uses one server
 app2 uses two servers
```

Creating a Policy Set Configuration: Example (cont)

2. Create every policy in a separate text file:

```
SERVER_POOL_NAMES=Free pool1 pool2
POLICY
 NAME=DayTime
 SERVERPOOL
   NAME=pool1
   IMPORTANCE=0
   MAX_SIZE=2
   MIN_SIZE=2
   SERVER_CATEGORY=
...
```

Creating a Policy Set Configuration: Example (cont)

3. Register the policies in a policy set

```
crsctl modify policyset -file <file_name>
```

4. Activate the required policy

```
crsctl modify policyset -attr "LAST_ACTIVATED_POLICY=DayTime"
```

Policy-managed Database Considerations

- Must be carefully planned and tested
  - Improper configuration may lead to making services unavailable
  - To check the effect of activating a policy without actually activating it:

```
crsctl eval activate policy NighShift -admin -l 'resources'
```

- More challenging troubleshooting
- Different instance naming convention
- Impact on GoldenGate
  - Clusterware Bundled Agents may be required

Summary

In this lecture, you should have learnt how to perform the following:

- Describe and create the following components in the Clusterware:
  - Server categorization
  - Cluster configuration policy
  - Cluster configuration policy set
- Obtain and set Server Configuration Attributes