ORACLE

Identity and Access Management

L100

Rohit Rahi
Oracle Cloud Infrastructure
Oct 2019

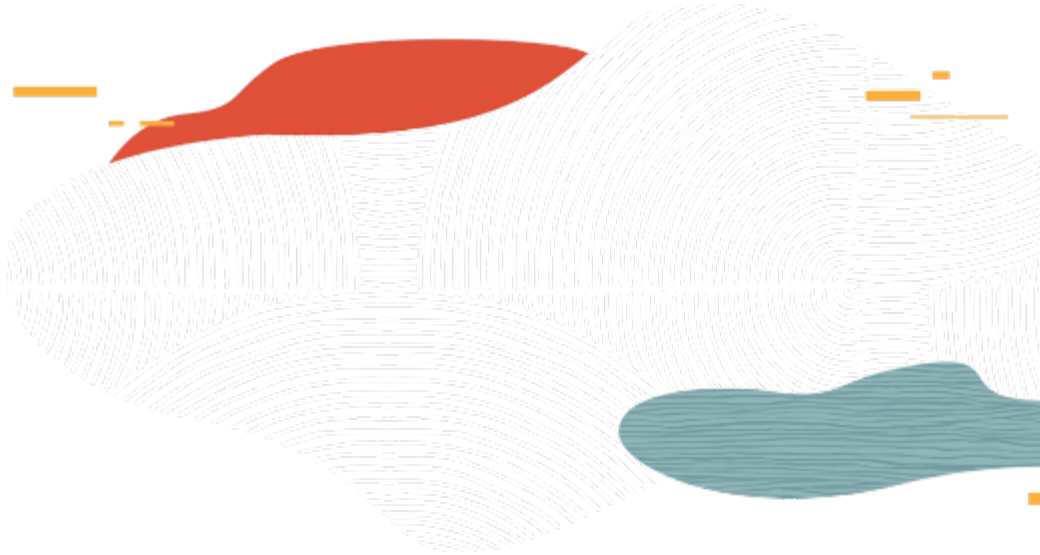
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Identity and Access Management

- Identity and Access Management (IAM) service enables you to control **what type of access** a **group of users** have and to **which specific resources**
- Resource is a cloud object that you create and use in OCI (e.g. compute instances, block storage volumes, Virtual Cloud Networks)
- Each OCI resource has a unique, Oracle-assigned identifier called an Oracle Cloud ID (OCID)
- IAM uses traditional identity concepts such as Principals, Users, Groups, AuthN, AuthZ and introduces a new capability called Compartment

ORACLE



Principals, AuthN, AuthZ

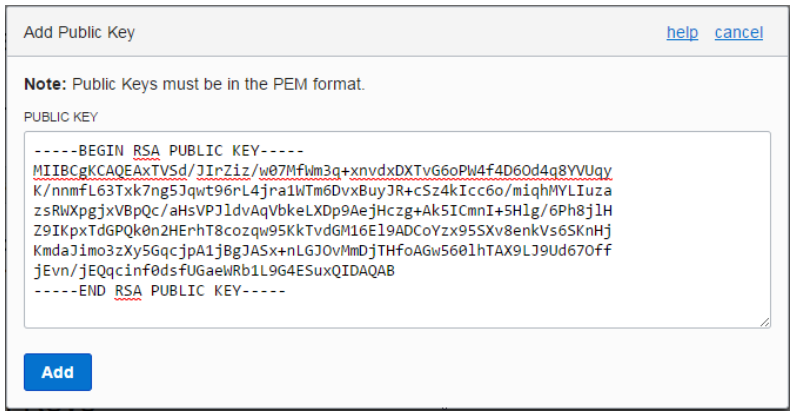
Principals

- A principal is an IAM entity that is allowed to interact with OCI resources
- Principals – IAM users and Instance Principals
- **IAM Users and Groups**
  - Users are persistent identities setup through IAM service to represent individual people or applications
  - When customers sign-up for an OCI account, the first IAM user is the default administrator
  - Default administrator sets up other IAM users and groups
  - Users enforce security principle of least privilege
    1. User has no permissions until placed in one (or more) groups and
    2. Group having at least one policy with permission to tenancy or a compartment
  - A Group is a collection of users who all need the same type of access to a particular set of resources
  - Same user can be member of multiple groups
- **Instance Principals**
  - Instance Principals lets instances (and applications) to make API calls against other OCI services removing the need to configure user credentials or a configuration file

Authentication

IAM service authenticates a Principal by –

- **User name, Password**
  - You use the password to sign in to the web console
  - An administrator will provide you with a one-time password when setting up your account
  - At your first log in, you are prompted to reset the password
- **API Signing Key**
  - Required when using the OCI API in conjunction with the SDK/CLI
  - Key is an RSA key pair in the PEM format (min 2048 bits)
  - In OCI Console, copy and paste the contents of the PEM public key file. Use the private key with the SDK or with your own client to sign your API requests
- **Auth Tokens**
  - Oracle-generated token strings to authenticate with 3rd party APIs that do no support OCI signature-based authentication (e.g. ADW)
  - Auth tokens do not expire

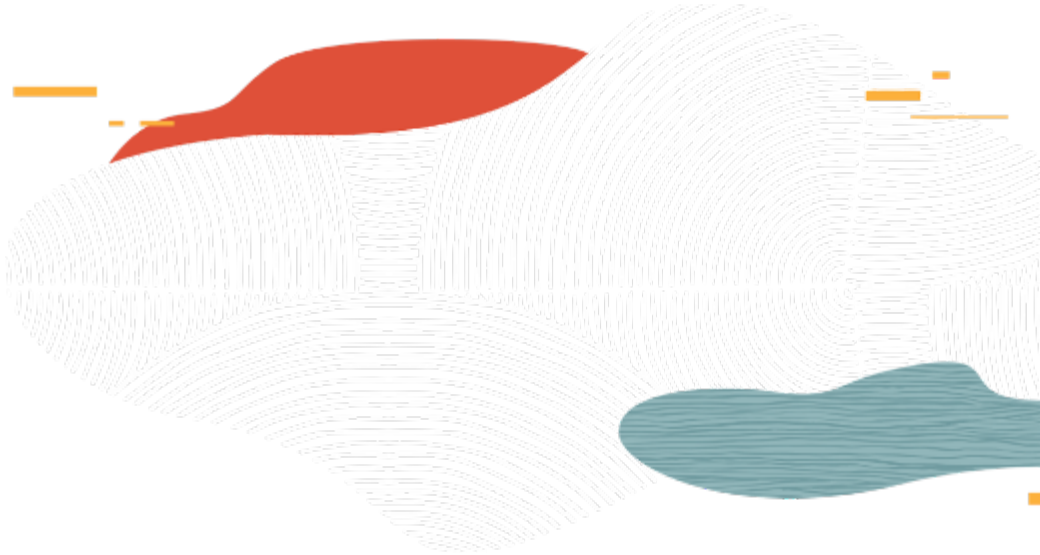



```
begin
 DBMS_CLOUD.create_credential (
     credential_name => 'OBJ_STORE_CRED',
     username => '<userXX>',
     password => '<your Auth Token>'
  ) ;
end;
/
```

Authorization

- Authorization specifies various actions an authenticated Principal can perform
- OCI Authorization - define specific privileges in policies and associating them with principals
- Supports security principle of least privilege; by default, users are not allowed to perform any actions (policies cannot be attached to users, but only groups)
- Policies are comprised of one or more statements which specify what groups can access what resources and at what level of access
- Policies are written in human-readable format:
  - Allow group <group_name> to <verb> <resource-type> in tenancy
  - Allow group <group_name> to <verb> <resource-type> in compartment <compartment_name> [where <conditions>]
- Policy Attachment: Policies can be attached to a compartment or the tenancy. Where you attach it controls who can then modify it or delete it

ORACLE



IAM Policies

Policy Syntax

Allow **<subject>** to **<verb>** **<resource-type>** in **<location>** where <conditions>

Verb	Type of access
inspect	Ability to list resources
read	Includes inspect + ability to get user-specified metadata/actual resource
use	Includes read + ability to work with existing resources (the actions vary by resource type)*
manage	Includes all permissions for the resource

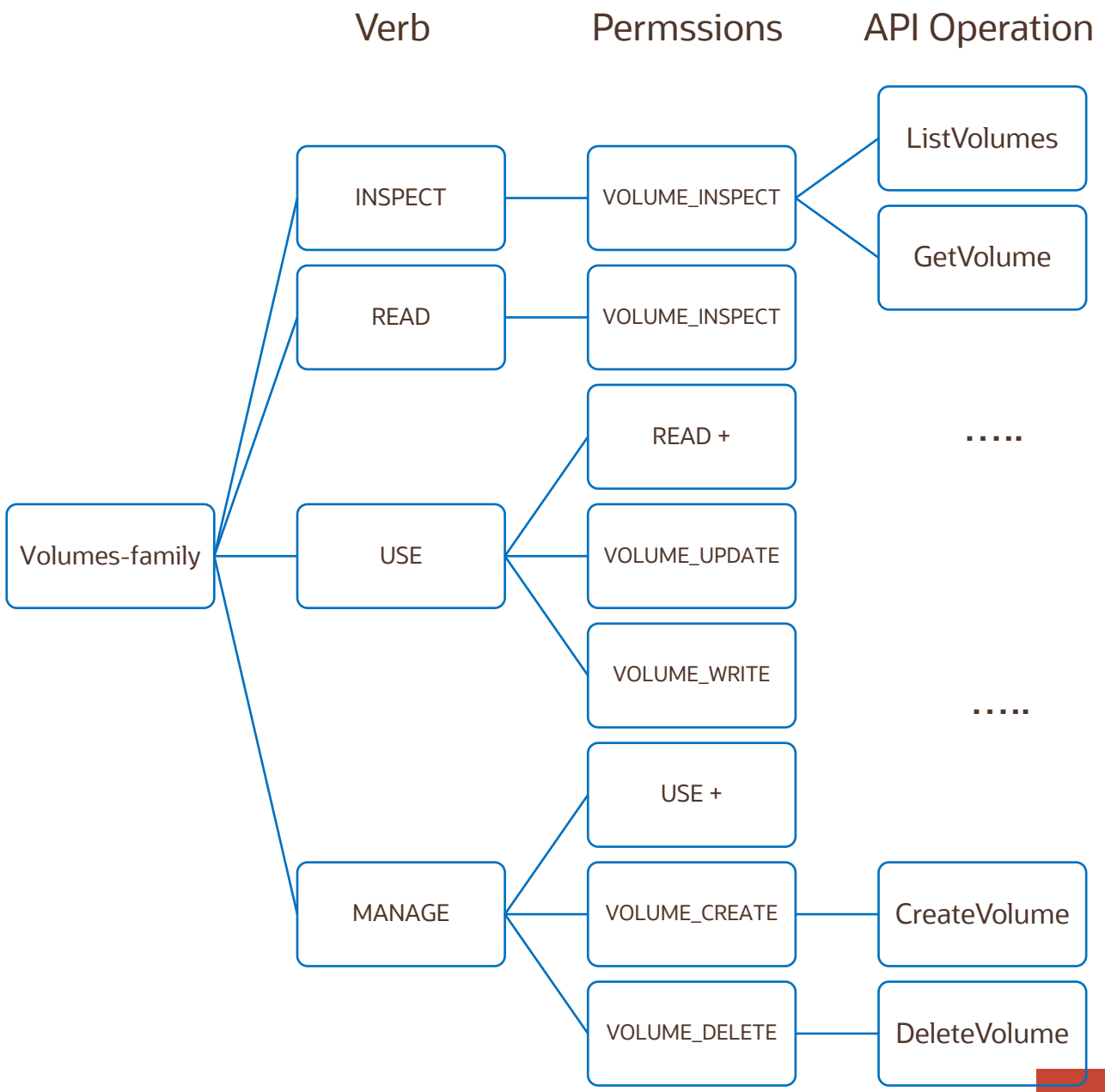
* In general, this verb does not include the ability to create or delete that type of resource

Aggregate resource-type	Individual resource type
all-resources	
database-family	db-systems, db-nodes, db-homes, databases
instance-family	instances, instance-images, volume-attachments, console-histories
object-family	buckets, objects
virtual-network-family	vcn, subnet, route-tables, security-lists, dhcp-options, and many more resources (link)
volume-family	volumes, volume-attachments, volume-backups
Cluster-family	clusters, cluster-node-pool, cluster-work-requests
File-family	file-systems, mount-targets, export-sets
dns	dns-zones, dns-records, dns-traffic,..

The IAM Service has no family resource-type, only individual ones

Verbs & Permissions

- When you write a policy giving a group access to a particular verb and resource-type, you're actually giving that group access to one or more predefined permissions
- Permissions are the atomic units of authorization that control a user's ability to perform operations on resources
- As you go from inspect > read > use > manage, the level of access generally increases, and the permissions granted are cumulative
- Each API operation requires the caller to have access to one or more permissions. E.g., to use ListVolumes or GetVolume, you must have access to a single permission: VOLUME_INSPECT



Common Policies

1. Network Admins manage a cloud network
    - Allow group NetworkAdmins to manage virtual-network-family in tenancy

2. Users launch compute instances
    - Allow group InstanceLaunchers to manage instance-family in compartment ABC

      Allow group InstanceLaunchers to read app-catalog-listing in tenancy

      Allow group InstanceLaunchers to use volume-family in compartment ABC

      Allow group InstanceLaunchers to use virtual-network-family in compartment XYZ

https://docs.cloud.oracle.com/iaas/Content/Identity/Concepts/commonpolicies.htm

ORACLE

Advanced IAM Policies

Advanced Policy Syntax

- As part of a policy statement, you can specify one or more *conditions* that must be met to get access

  Allow <subject> to <verb> <resource-type> in <location> where **<conditions>**

- You use variables when adding conditions to a policy; 2 types
  - **request** – relevant to the request itself
  - **target** – relevant to the resource(s) being acted upon in the request)
  - E.g. variable request.operation represents the API operation being requested (e.g. ListUsers); target.group.name represents the name of the group
- variable name is prefixed accordingly with either request or target followed by a period
- Examples:
  - Allow group Phoenix-Admins to manage all-resources in tenancy where request.region='phx'

https://docs.cloud.oracle.com/iaas/Content/Identity/Reference/policyreference.htm#Resource

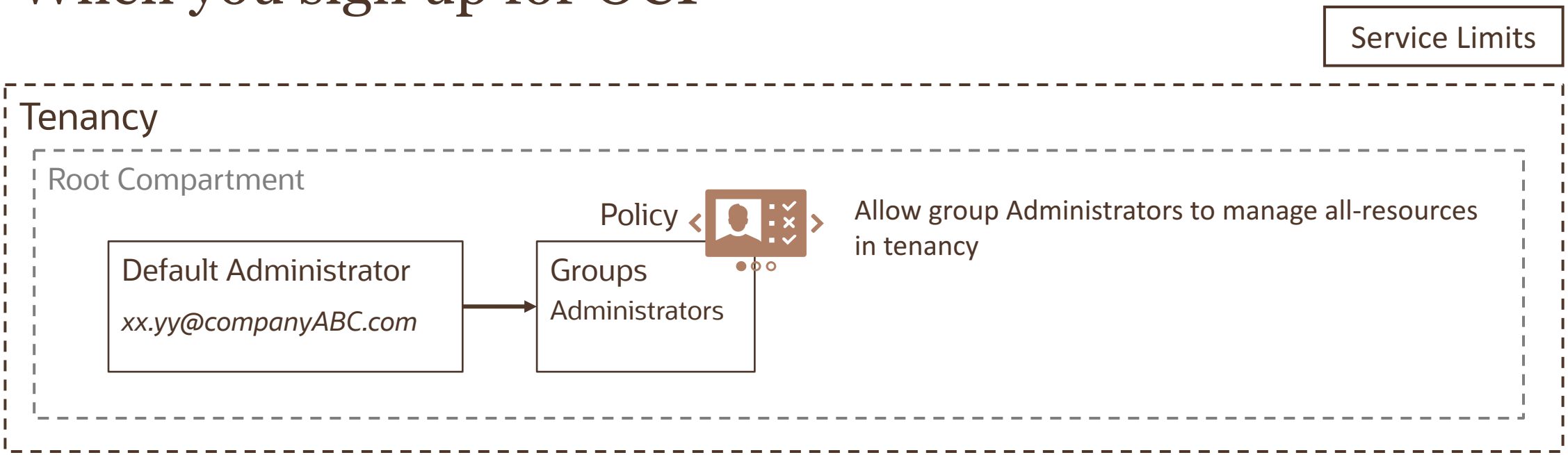
ORACLE

Compartments

Compartment

- A compartment is a collection of related resources (VCN, instances,..) that can be accessed only by groups that have been given permission (by an administrator in your organization)
- Compartments help you organize and control access to your resources
- Design considerations:
  - Each resource belongs to a single compartment but resources can be connected/shared across compartments (VCN and its subnets can live in different compartments)
  - A compartment can be deleted after creation or renamed
  - A compartment can have sub compartments that can be up to six levels deep
  - Most resources can be moved to a different compartment after they are created (some restrictions apply)
  - After creating a compartment, you need to write at least one policy for it, otherwise it cannot be accessed (except by administrators or users who have permission to the tenancy)
  - Sub compartment inherits access permissions from compartments higher up its hierarchy
  - When you create a policy, you need to specify which compartment to attach it to

When you sign up for OCI



- Oracle sets up a default administrator for the account
- Default Group Administrators
  - Cannot be deleted and there must always be at least one user in it
  - Any other users placed in the Administrators group will have full access to all of resources
  - Tenancy Policy gives Administrators group access to all resources – this policy can't be deleted/changed

- Root Compartment can hold all the cloud resources
- Best practice is to create dedicated Compartments when you need to isolate resources

ORACLE

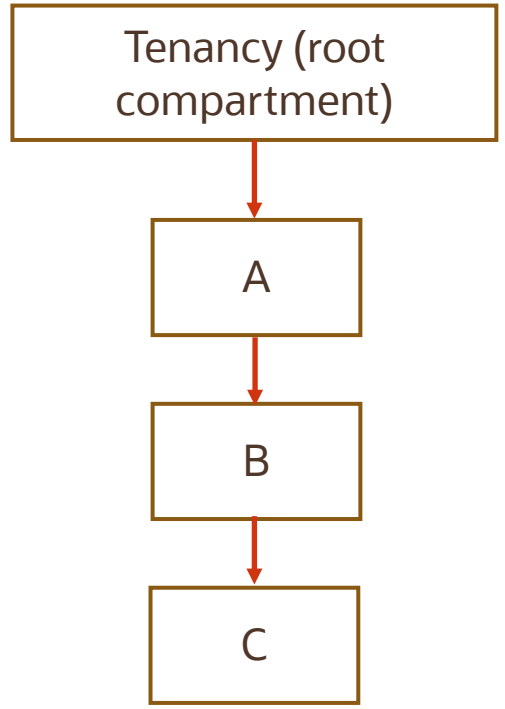
Policy Inheritance and Attachment for Compartments

Policy Inheritance

- Concept of inheritance: Compartments inherit any policies from their parent compartment
  - E.g. OCI has a built-in policy for Administrators, Allow group Administrators to manage all-resources in tenancy
  - Due to Policy Inheritance, the Administrators group can also do anything in any of the compartments in the tenancy

- Three levels of compartments: A, B, and C
  - Policies that apply to resources in Compartment A also apply to resources in Compartments B and C

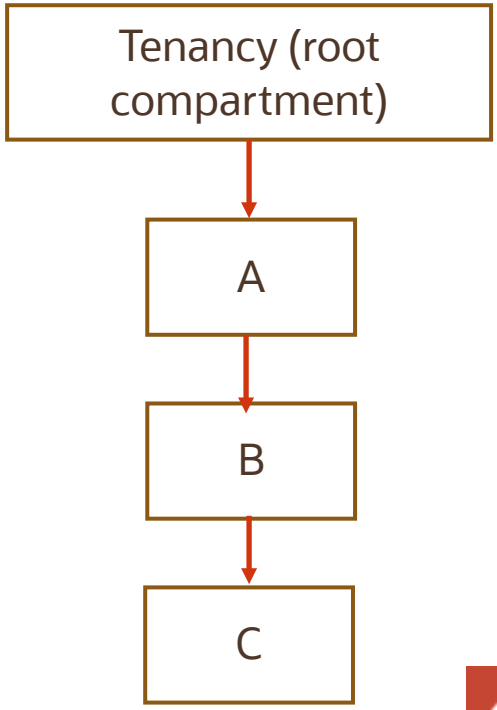
  - Allow group NewtworkAdmins to manage virtual-network-family in compartment A allows the group NetworkAdmins to manage VCNs in Compartment A, B, and C



Policy Attachment

- Concept of attachment: when you create a policy you must attach it to a compartment (or tenancy). Where you attach it controls who can then modify it or delete it
  - Attach it to tenancy (root compartment), then anyone with access to manage policies in the tenancy can then change or delete it.
  - Attach to a child compartment, then anyone with access to manage the policies in that compartment (e.g. compartment admins) can change or delete it

- You want to create a policy to allow NetworkAdmins to manage VCNs in Compartment C. Attach to
  - C or B – Allow group NewtworkAdmins to manage virtual-network-family in compartment C
  - A – Allow group NewtworkAdmins to manage virtual-network-family in compartment B:C
    - Only Compartment A admins can modify it
    - NetworkAdmins can still only manage VCNs in CompartmentC
  - Tenancy – Allow group NewtworkAdmins to manage virtual-network-family in compartment A:B:C



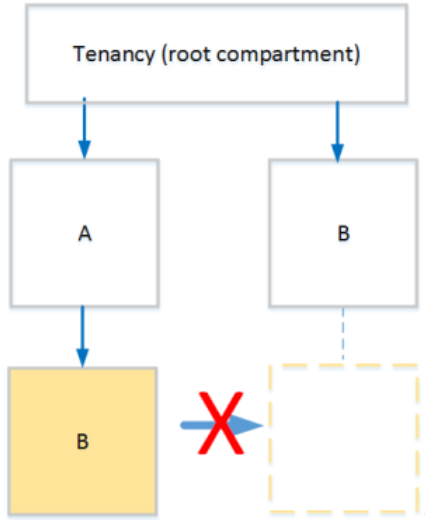
ORACLE

Moving Compartments

Moving a Compartment to a different Parent Compartment

- You can move a compartment to a different parent compartment within the same tenancy. When you move a compartment, all its contents (sub compartments and resources) are moved with it.

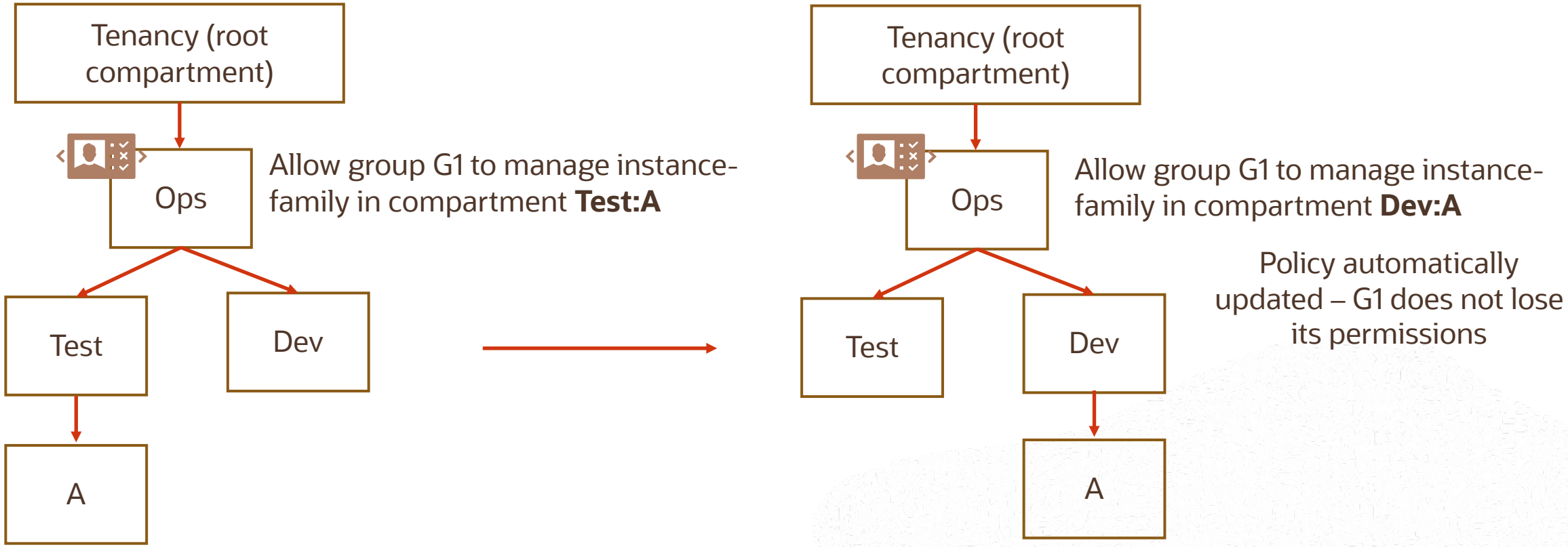
- Restrictions:
  - You can't move a compartment to a destination compartment with the same name as the compartment being moved
  - Two compartments within the same parent cannot have the same name. Therefore you can't move a compartment to a destination compartment where a compartment with the same name already exists



You can’t move a compartment under a parent compartment with the same name

Policy Implications when moving compartments

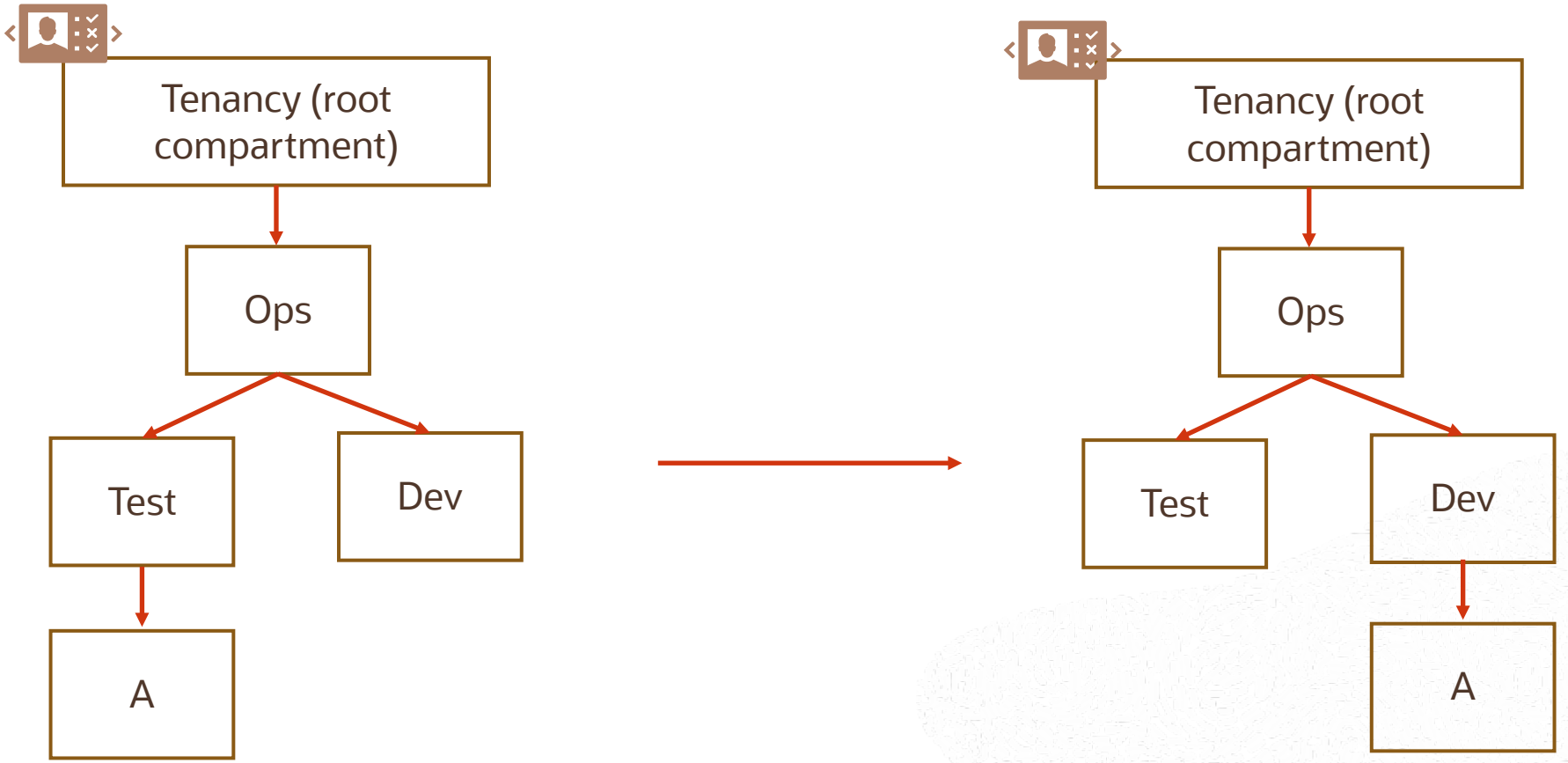
Policies that specify the **compartment hierarchy down to the compartment being moved** will automatically be updated when the policy is attached **to a shared ancestor of the current and target parent**



Policy Implications when moving compartments

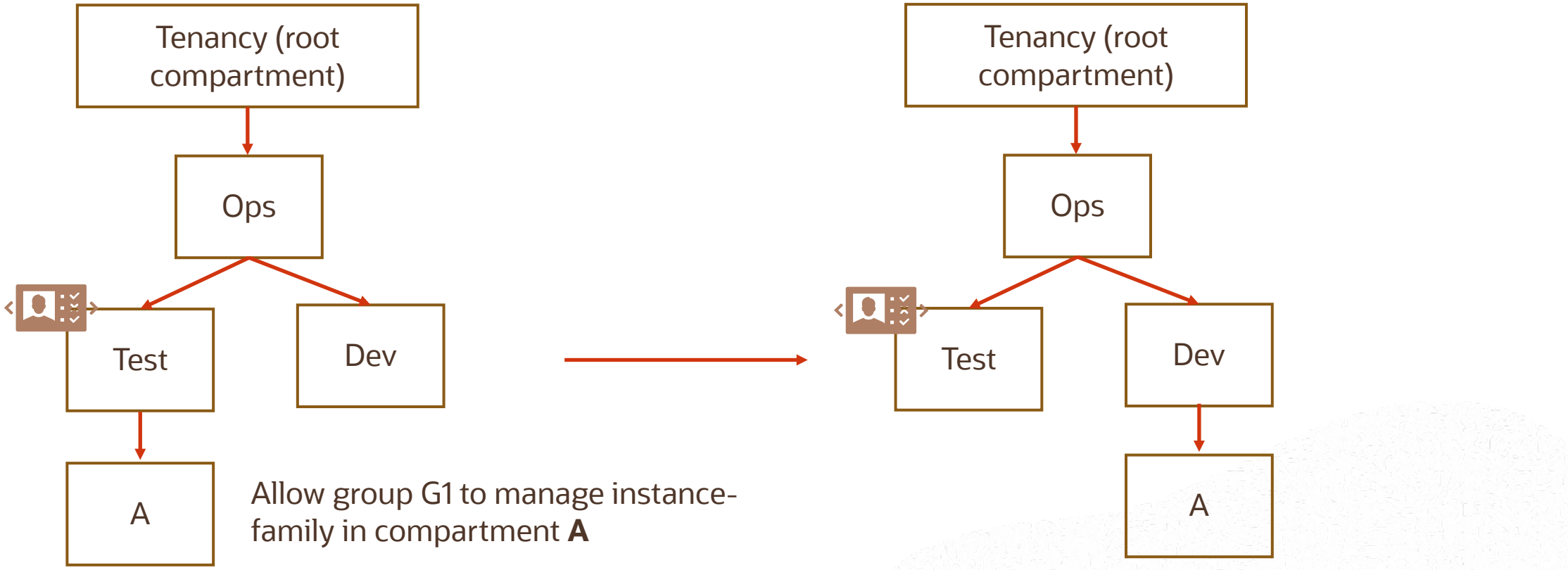
Allow group G1 to manage instance-family in compartment **Ops:Test**
Allow group G2 to manage instance-family in compartment **Ops:Dev**

G1 can no longer manage instances in compartment A
G2 can now manage instances in compartment A



Policy Implications when moving compartments

Policy attached directly to a compartment moved is not automatically updated



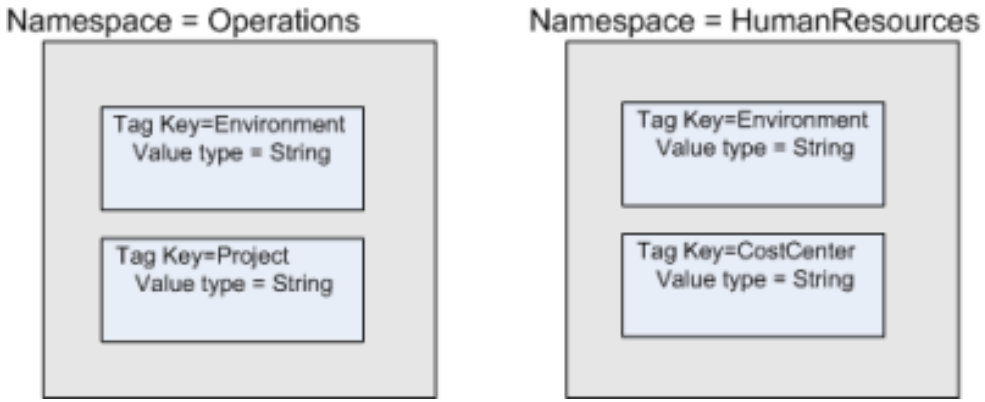
ORACLE

Tags

Tagging

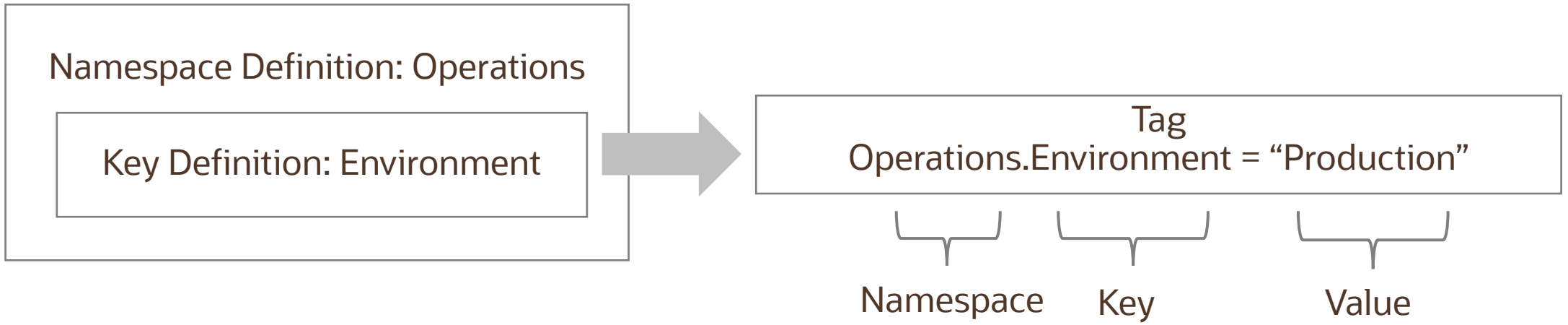
- Free-form Tags – basic implementation
  - Consist simply of a key and a value

- Defined Tags – more features and control
  - Are contained in tag Namespaces
  - Defined schema, secured with Policy



Tag Namespace

- A Tag Namespace is a container for set of tag keys with tag key definitions
- Tag key definition specifies its key (environment) and what types of values are allowed (string, number, text, date, enumerations, etc.)

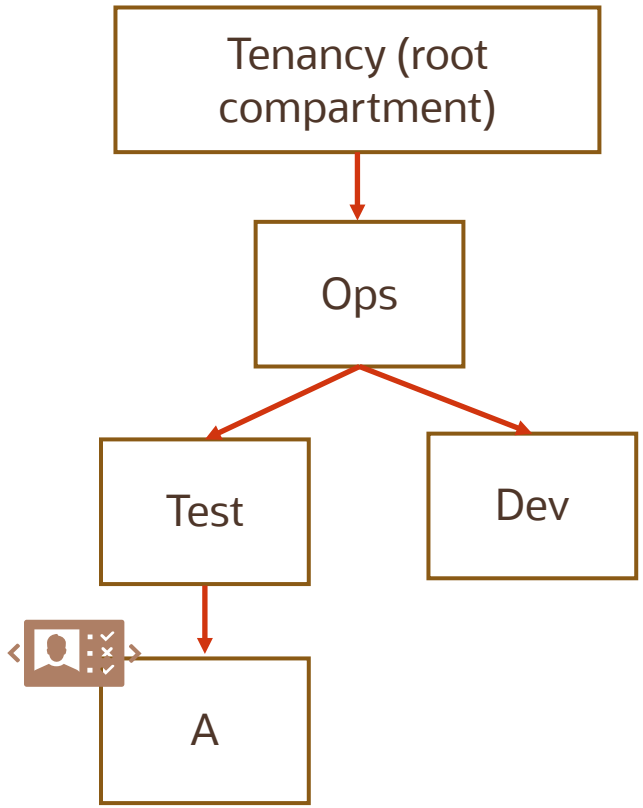


- Tag key definition or a tag namespace cannot be deleted, but retired. Retired tag namespaces and key definitions can no longer be applied to resources
- You can reactivate a tag namespace or tag key definition that has been retired to reinstate its usage in your tenancy

Working with defined tags

- Consist of a tag namespace, a key, and a value
- Tag namespace and tag key definition must be set up in your tenancy before users can apply them
- A tag key can have either a tag value type of string or a list of values (from which the user must choose)
- You can use a variable to set the value of a tag. When you add the tag to a resource, the variable resolves to the data it represent. E.g.
  - Operations.CostCenter = ${iam.principal.name} at ${oci.datetime}
  - Operations is the namespace, CostCenter is the tag key, and the tag value contains two tag variables ${iam.principal.name} and ${oci.datetime}
  - When you add this tag to a resource, the variable resolves to your user name (the name of the principal that applied the tag) and a time date stamp for when you added the tag

Defined tags work with Policies



Allow group InstanceLaunchers to manage instance-family in compartment A

Allow group InstanceLaunchers to use volume-family in compartment A

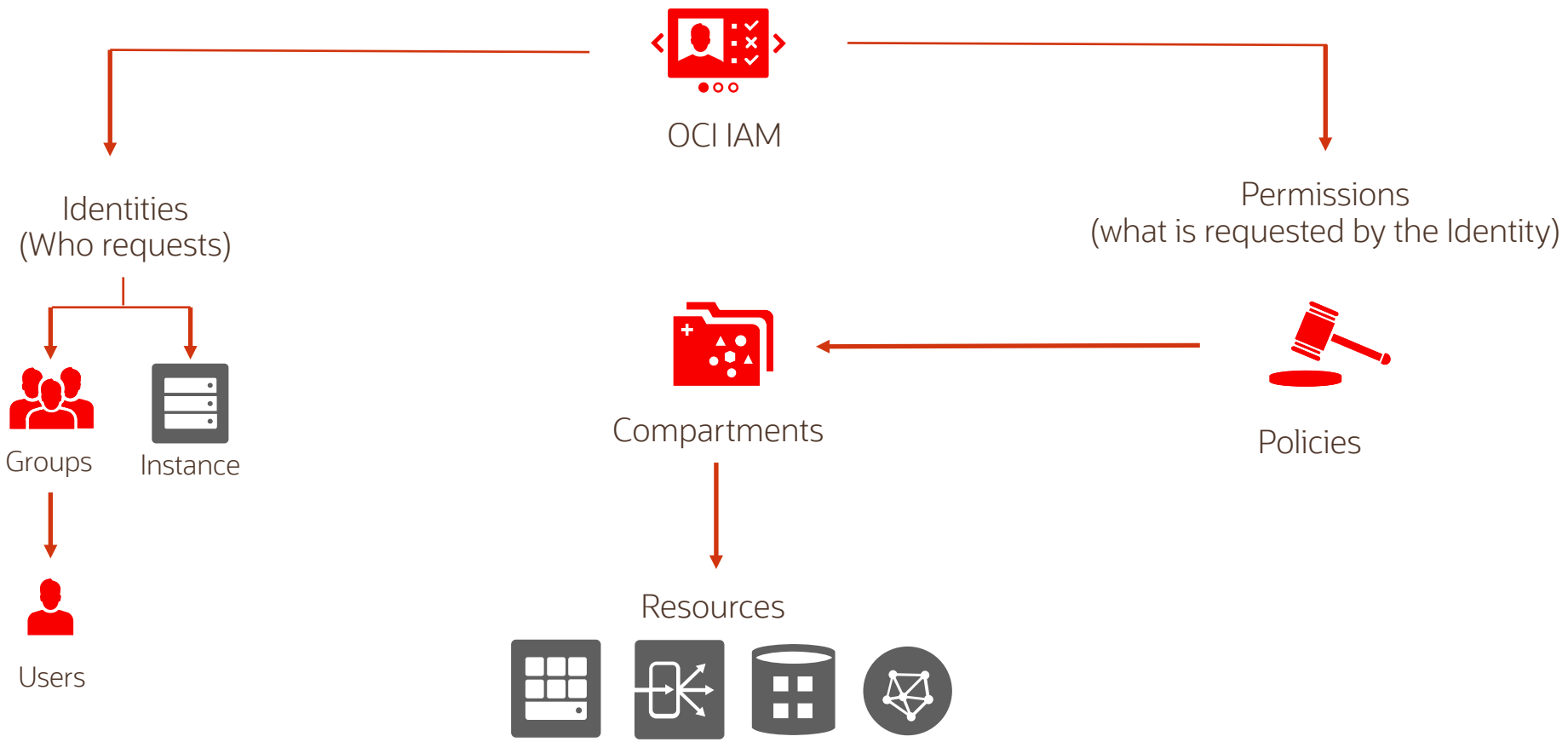
Allow group InstanceLaunchers to use virtual-network-family in compartment A

Allow group InstanceLaunchers to use tag-namespaces in compartment A where target.tag-namespace.name='Operations'

Users in the InstanceLaunchers group can now apply the Operations.CostCenter tag to resources in Compartment A

Summary

Identity and Access Management (IAM) service enables you to control what type of access a group of users have and to which specific resources



Summary

- IAM Principals – IAM users and Instance Principals
- Authentication – username/password, API Signing keys, Auth Tokens
- Authorization – Policies and associating them with Principals
- Policies syntax and examples of advanced policies
- Compartment, a unique OCI feature, can be used to organize and isolate related cloud resources
- Concept of Policy Inheritance and Attachment for compartments
- OCI supports both free form tags and defined tags with a schema and secured by policies

ORACLE

Oracle Cloud always free tier:
oracle.com/cloud/free/

OCI training and certification:
cloud.oracle.com/en_US/iaas/training
cloud.oracle.com/en_US/iaas/training/certification
education.oracle.com/oracle-certification-path/pFamily_647

OCI hands-on labs:
ocitraining.qloudable.com/provider/oracle

Oracle learning library videos on YouTube:
youtube.com/user/OracleLearning

Thank you