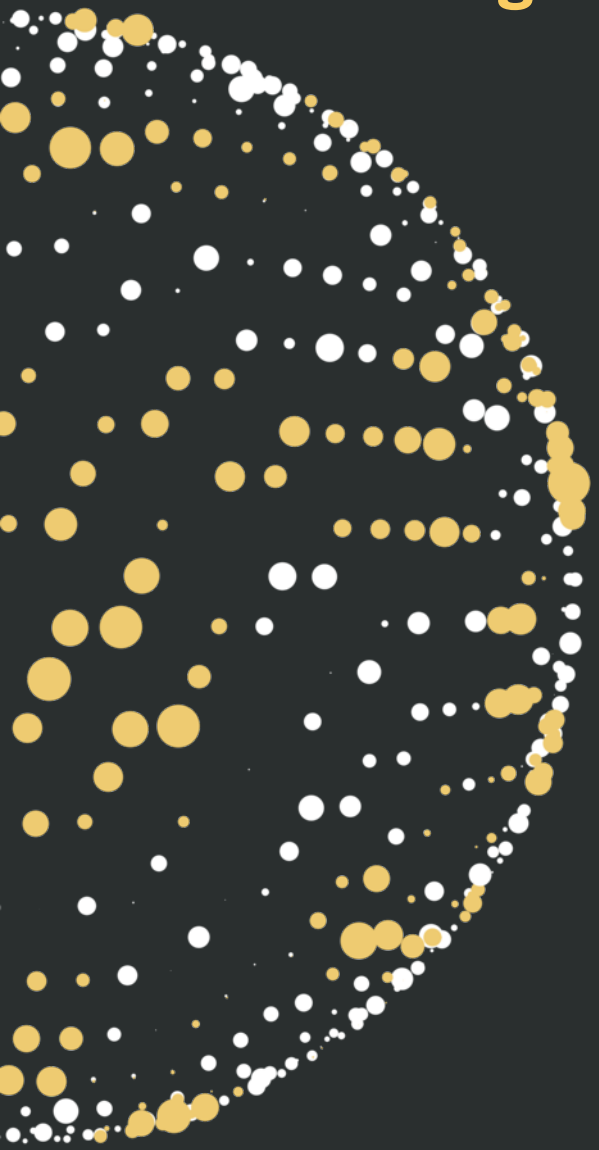
Identity & Access Mgmt

Who has access to your cloud resources?

Alexandre Fagundes
LAD Partner Enablement

Agenda



- IAM: What is?
- How it works
- Identity Concepts
- Policies syntax and advanced policies
- Compartments
- Concept of Policy Inheritance and Attachment
- Free form tags, defined tags and TBAC
- Understand Multi-Factor Authentication (MFA)
- Users Federation
- Design reference IAM Model for an Enterprise
- Real life story for IAM compartment and policy design

Identity and Access Management (IAM) service

Controls what **type of access** a **Group of users** have and to…

…which specific resources

What is OCI IAM?

- **Fine-Grained Access Control** — Who can access certain data?
- **AuthN** — Who are you?
- **AuthZ** — What permissions do you have?
- **Auditing** — What did you do?

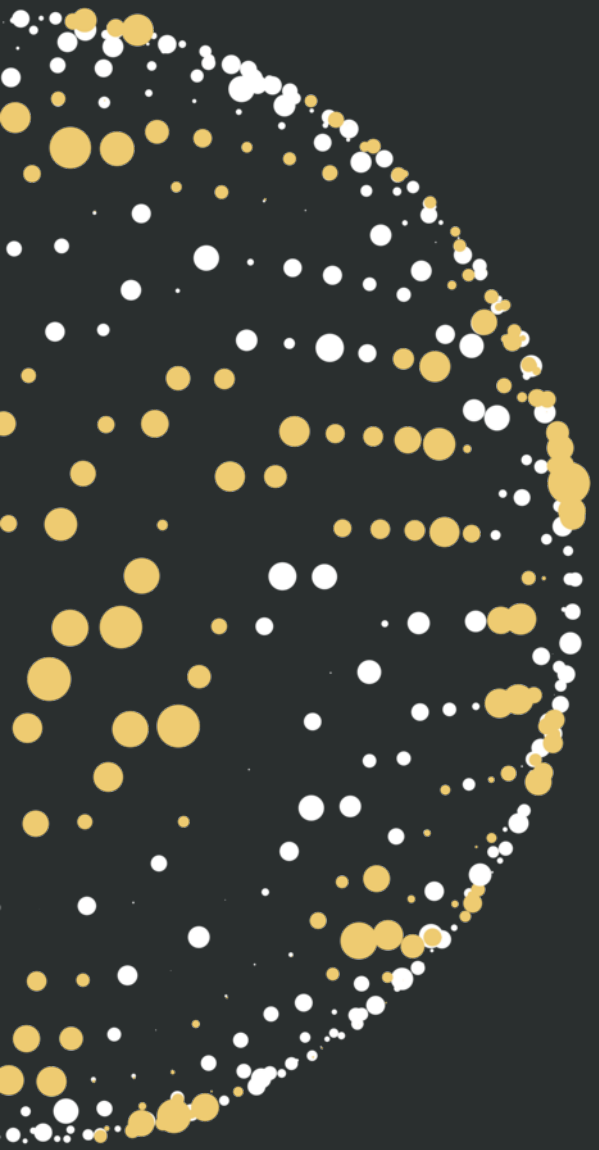
How it works?

IAM uses traditional identity concepts such as Principals, Users, Groups, Dynamic Groups, AuthN, AuthZ and introduces a new capability called Compartment

Resource is nothing but a cloud object!

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)



Identity Concepts

Principals

A principal is an IAM entity like users and Instance Principals that is allowed to interact with OCI resources

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

IAM Users, Groups, Dynamic 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

- User has no permissions until placed in one (or more) groups and
- Group having at least one policy with permission to tenancy or a compartment

Group is a collection of users who all need the same type of access to a particular set of resources

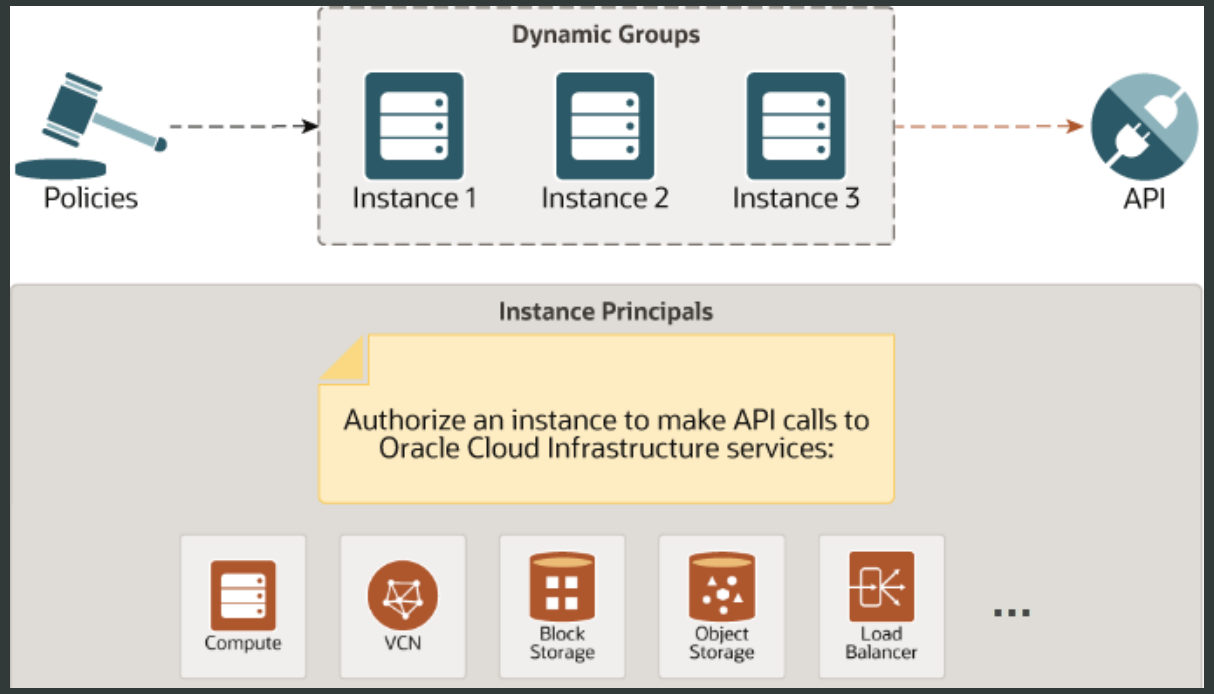
Same user can be member of multiple groups

Dynamic Groups

Allows principals to be grouped as “principal actors” (similar to a group of users)

Membership changes (e.g., scale instances up & down)

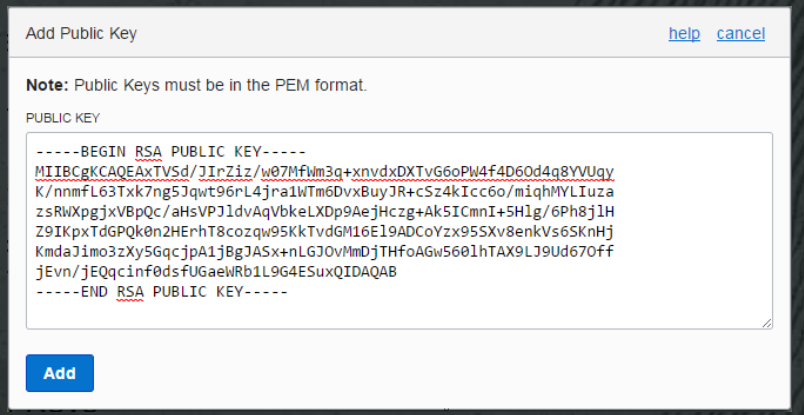
Dynamic Groups relies on Matching Rules



Authentication

IAM service authenticates a Principal by –

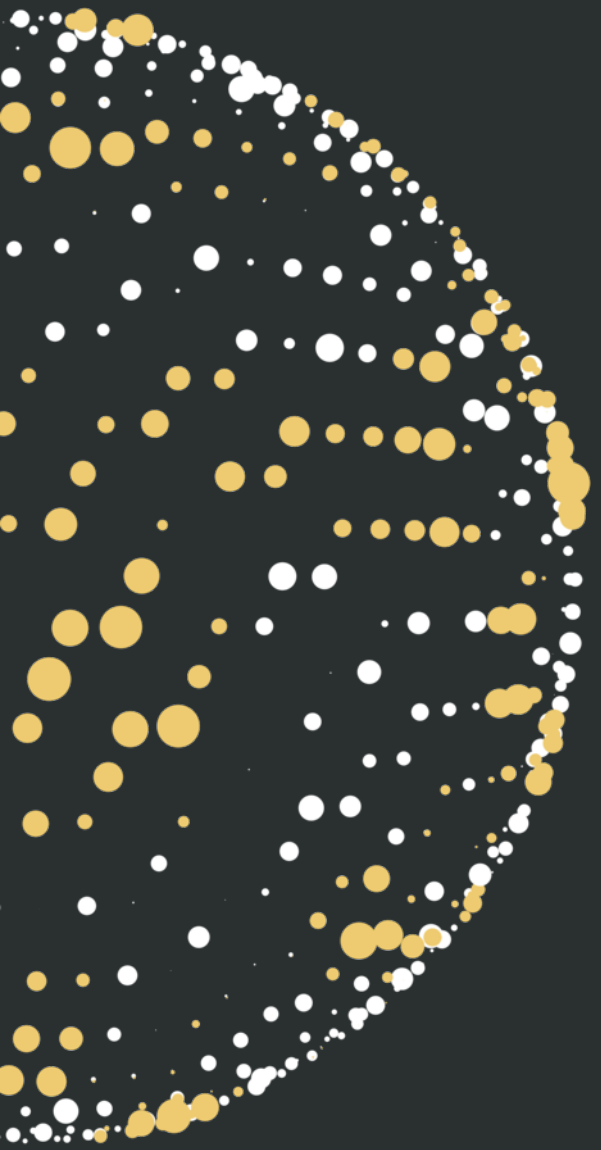
- **User name, Password**
  - You use the password to sign in to the web console
  - An administrator will provide you with a one-time password when setting up your account
  - At your first log in, you are prompted to reset the password
- **API Signing Key**
  - Required when using the OCI API in conjunction 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 <action> <resource-type> in tenancy
  - Allow group <group_name> to <action> <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



IAM Policies

Policy Syntax

allow <subject> to <actions> <resource-type> in <placement> where <conditions>

SUBJECT	Type of Access
Group	Both ID & Name
Dynamic Group	Both ID & Name
Service	Only Name

Subjects are a clause for the various ways that an authenticated actor can be addressed:

By membership in a group (e.g., “group Admins”, “group id ocid1.group.c1.....”)

As a service, for OCI Services (e.g., “service objectstorage-br-saopaulo-1”)

As a wildcard, with “Any User” (any request from the tenancy)

More than one name or group can be named in Subjects element. These can be chained by kind (e.g., “group Admins, Readers”)

Policy Syntax

Allow **<subject>** to **<actions>** **<resource-type>** in **<placement>** where <conditions>

Action	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

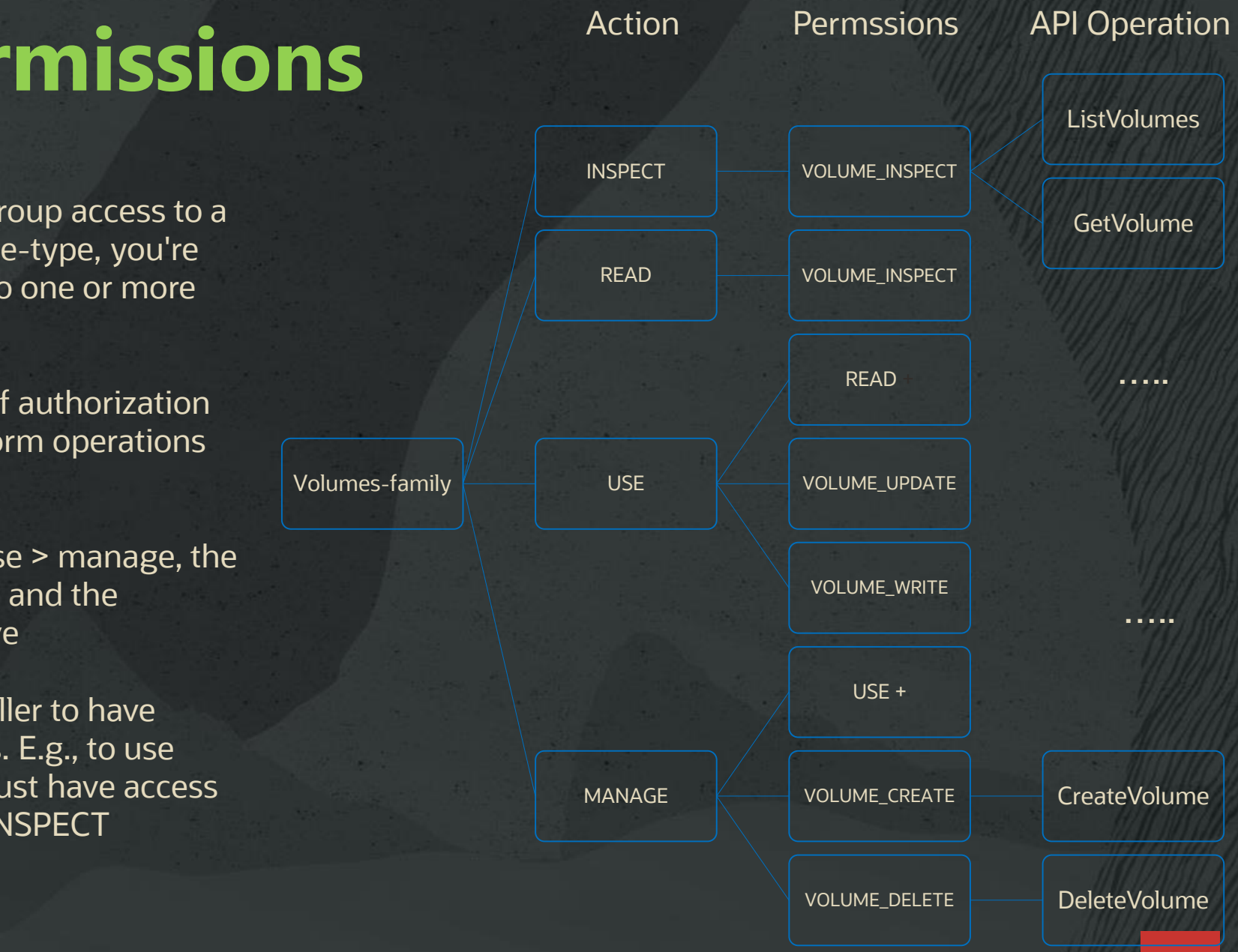
* In general, this verb does not include the ability to create or delete that type of resource

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,..

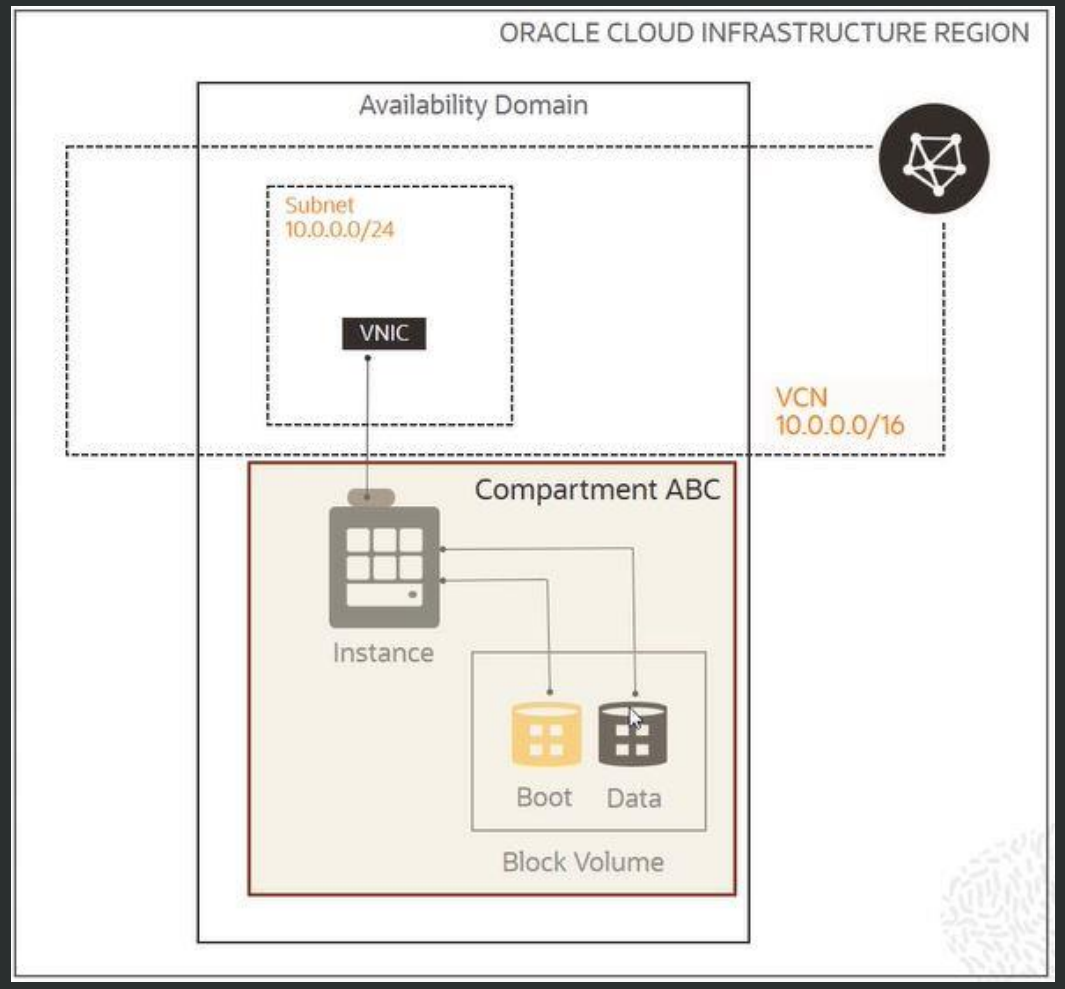
The IAM Service has no family resource-type, only individual ones

Actions & Permissions

- When you write a policy giving a group access to a particular action(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



Network Admins manage a cloud network

- Allow group NetworkAdmins to manage virtual-network-family in tenancy

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 tenancy

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

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)

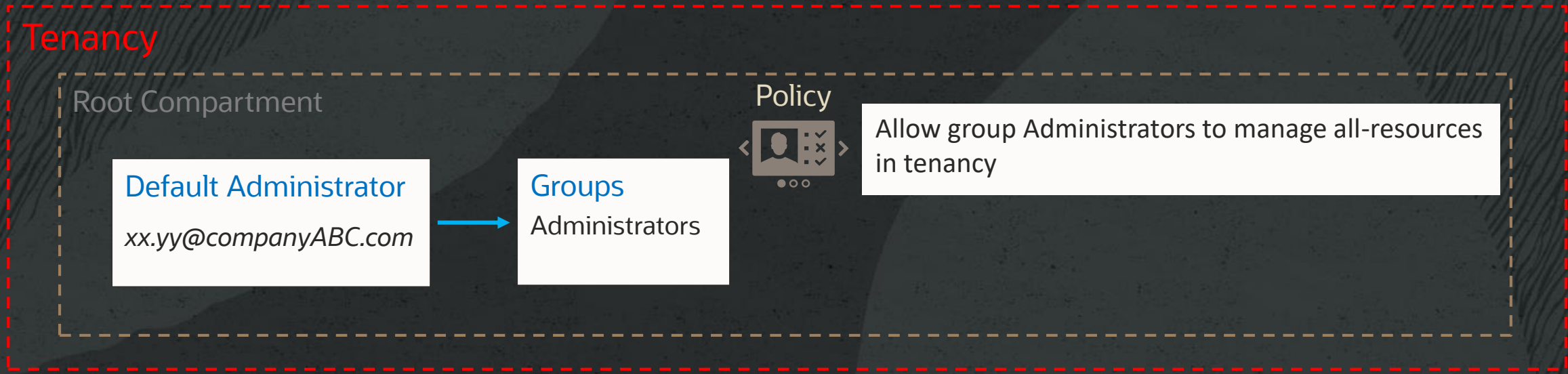
Compartments help you organize and control access to your resources

Few 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

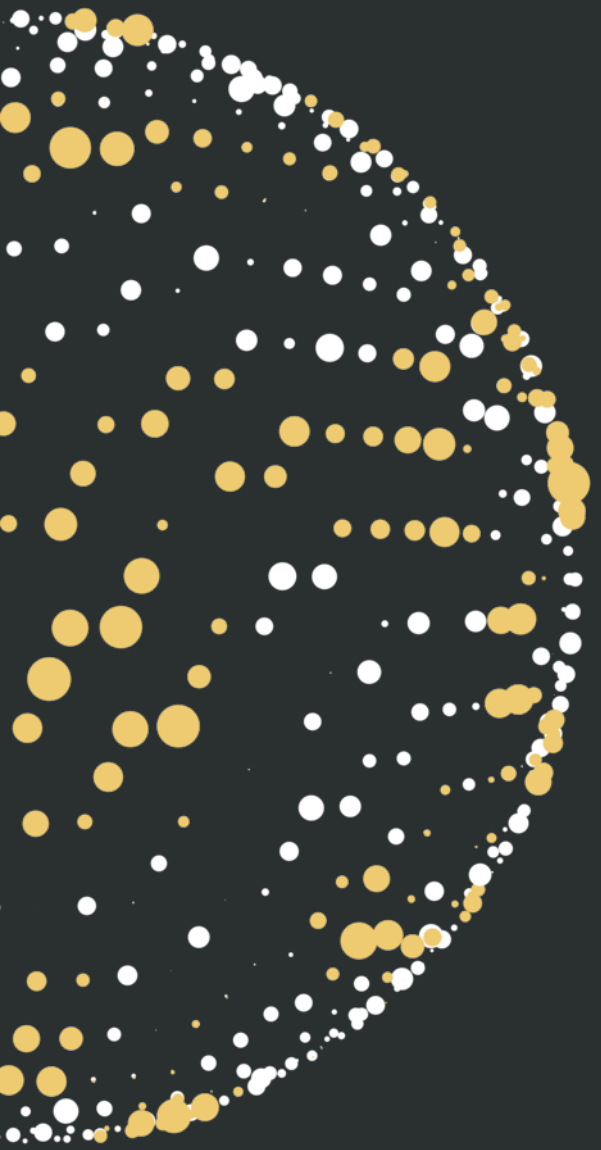
When you sign up for OCI

Service Limits



- 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

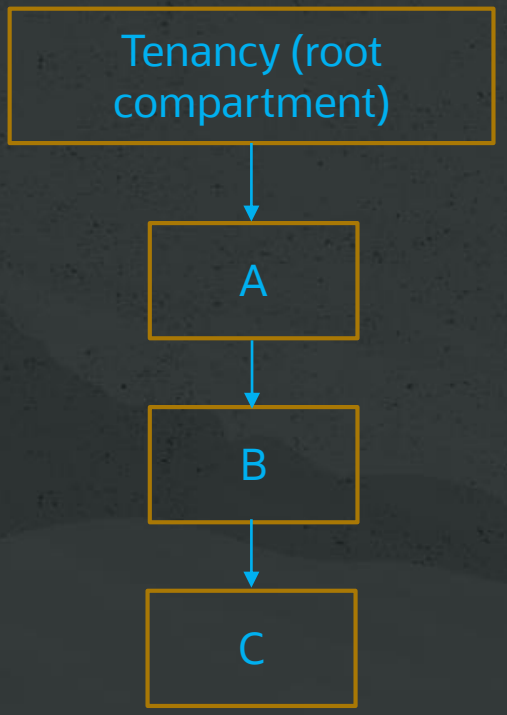
- Root Compartment can hold all the cloud resources
- Best practice is to create dedicated Compartments when you need to isolate resources



Policy Inheritance and Attachment for Compartments

Policy Inheritance

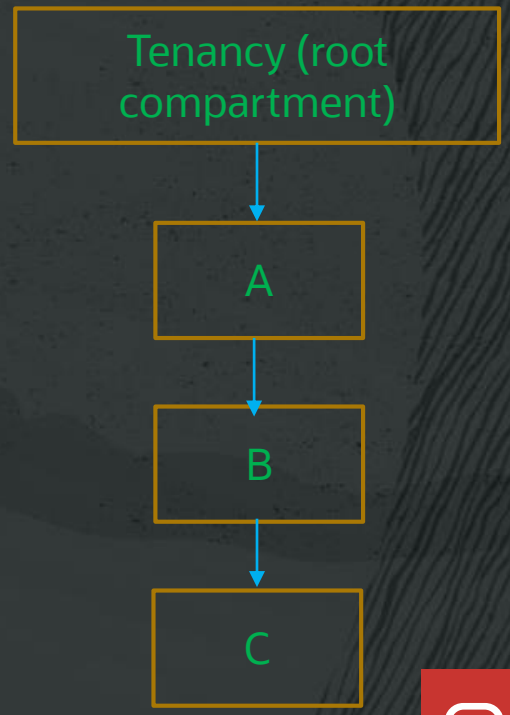
- Concept of inheritance: Compartments inherit any policies from their parent compartment
  - E.g. OCI has a built-in policy for Administrators, Allow group Administrators to manage all-resources in tenancy
  - Due to Policy Inheritance, the Administrators group can also do anything in any of the compartments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

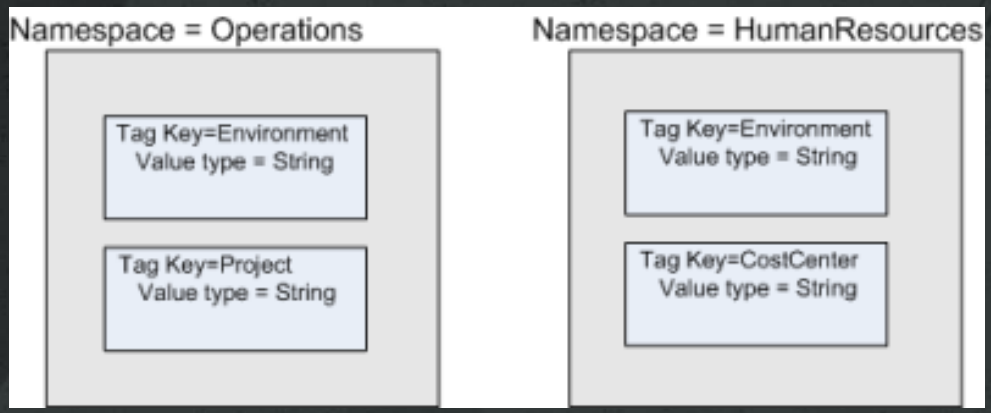


Tags

Tagging

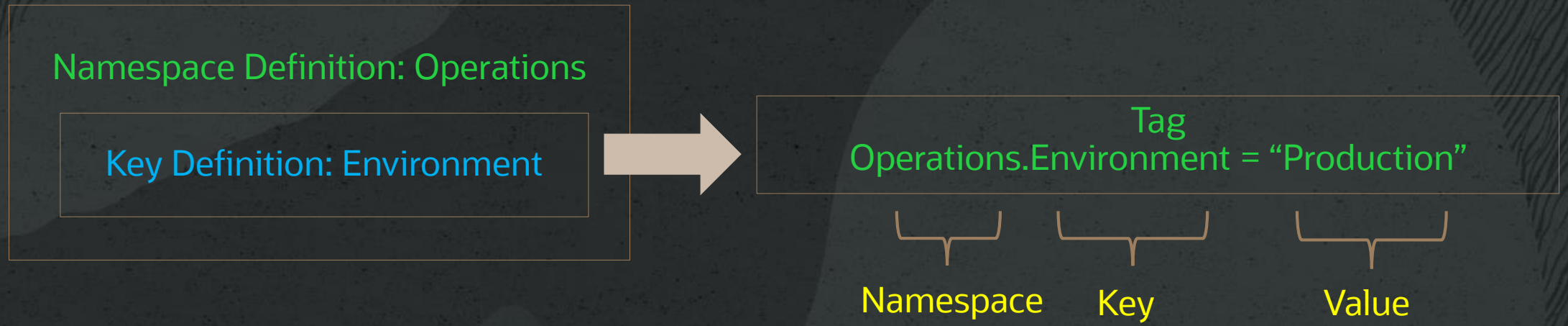
- Free-form Tags – basic implementation
  - Consist simply of a key and a value

- 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

Tag-based Access Control

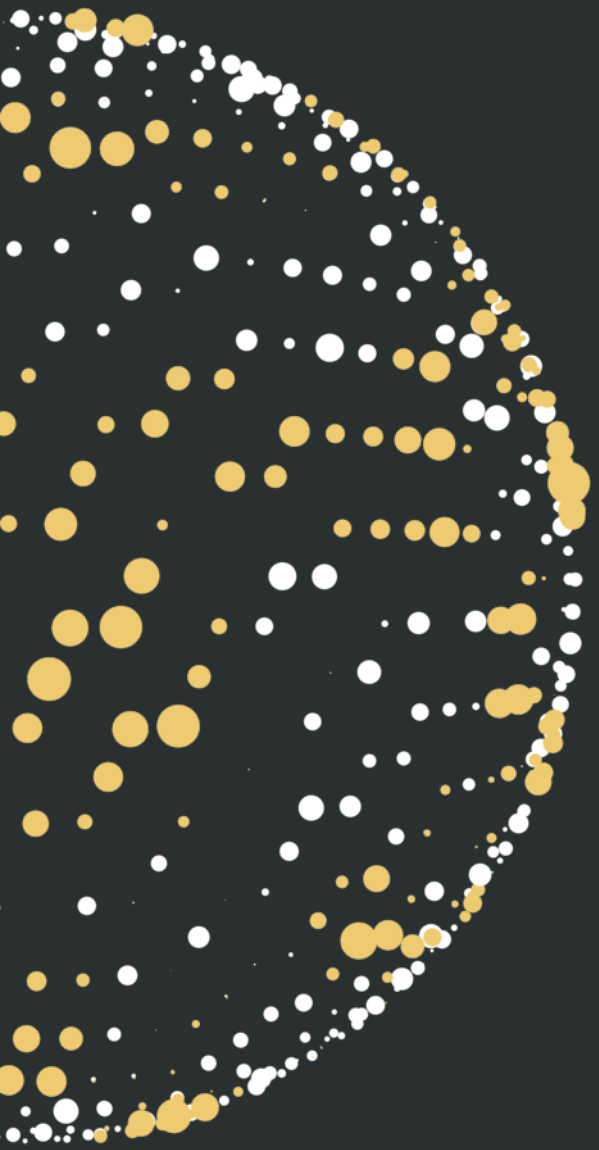
- TBAC Tag-based Access Control allows to define policies with tags that span compartments, groups and resources
- Scope access based on the tags applied to a resource
- TBAC = conditions + set of tag variables
- Access can be controlled based on a tag
  - On the requesting resource (group, dynamic group or compartment)
  - On the target of the request (resource or compartment

Tag-based Access Control

Tag applied to requestor	Variable	Sample Policy
Group	request.principal.group.tag.{tagNamespace}.{tagKeyDefinition}='<value>'	allow any-user to manage instances in compartment HR where request.principal.group.tag.Operations.Projetct='PROD'
Dynamic Group	request.principal.group.tag.{tagNamespace}.{tagKeyDefinition}='<value>'	allow dynamic-group InstancesA to manage object-family in compartment HR where request.principal.group.tag.Operations.Project='PROD'
Compartment	request.principal.compartment.tag.{tagNamespace}.{tagKeyDefinition}='<value>'	allow dynamic-group InstancesA to manage object-family in compartment HR where request.principal.compartment.tag.Operations.Project='PROD'

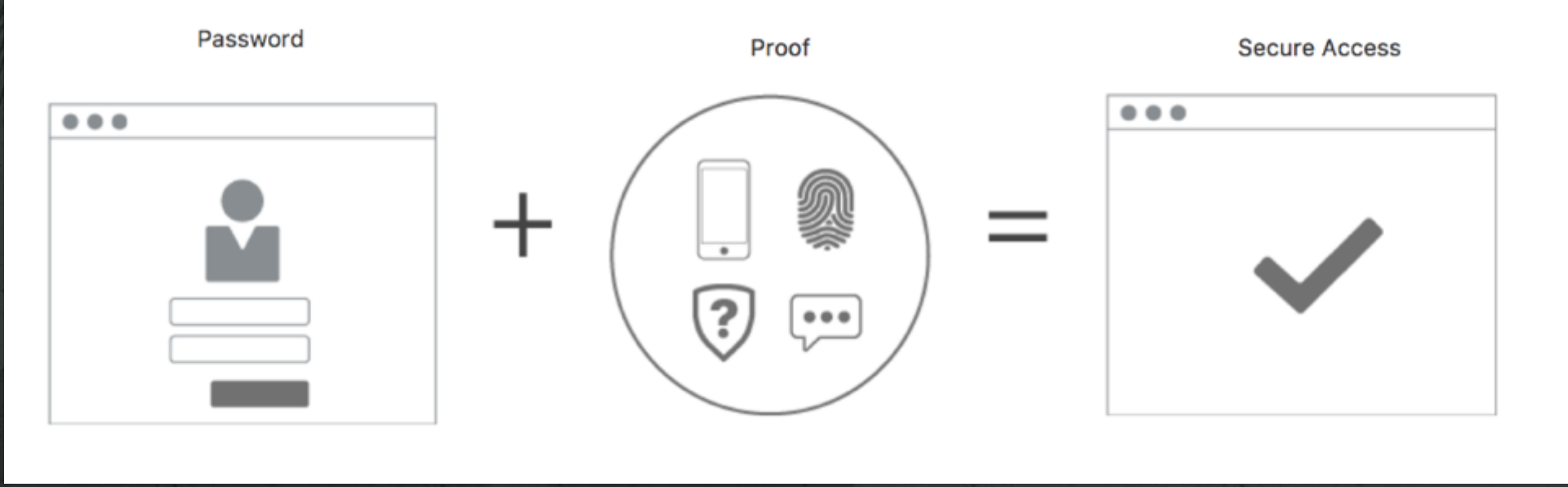
Tag-based Access Control

Tag applied to target	Variable	Sample Policy
Resource	target.resource.tag.{tagNamespace}.{tagKeyDefinition}='<value>'	allow group groupA to manage all-resources in compartment HR where target.resource.tag.Operations.Project='PRD'
Compartment	target.resource.compartment.tag.{tagNamespace}.{tagKeyDefinition}='<value>'	allow group groupA to manage all-resources in compartment HR where target.resource.compartment.tag.Operations.Project='PRD'



Multifactor Authentication (MFA)

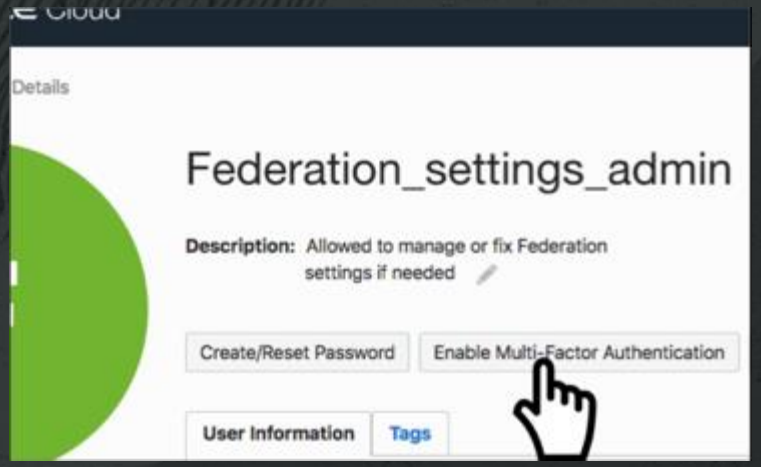
Multi-factor Authentication (MFA) - General Concepts



Multi-factor authentication (MFA) is a method of authentication that requires the use of more than one factor to verify a user's identity. Examples of authentication factors are a password (something you know) and a device (something you have).

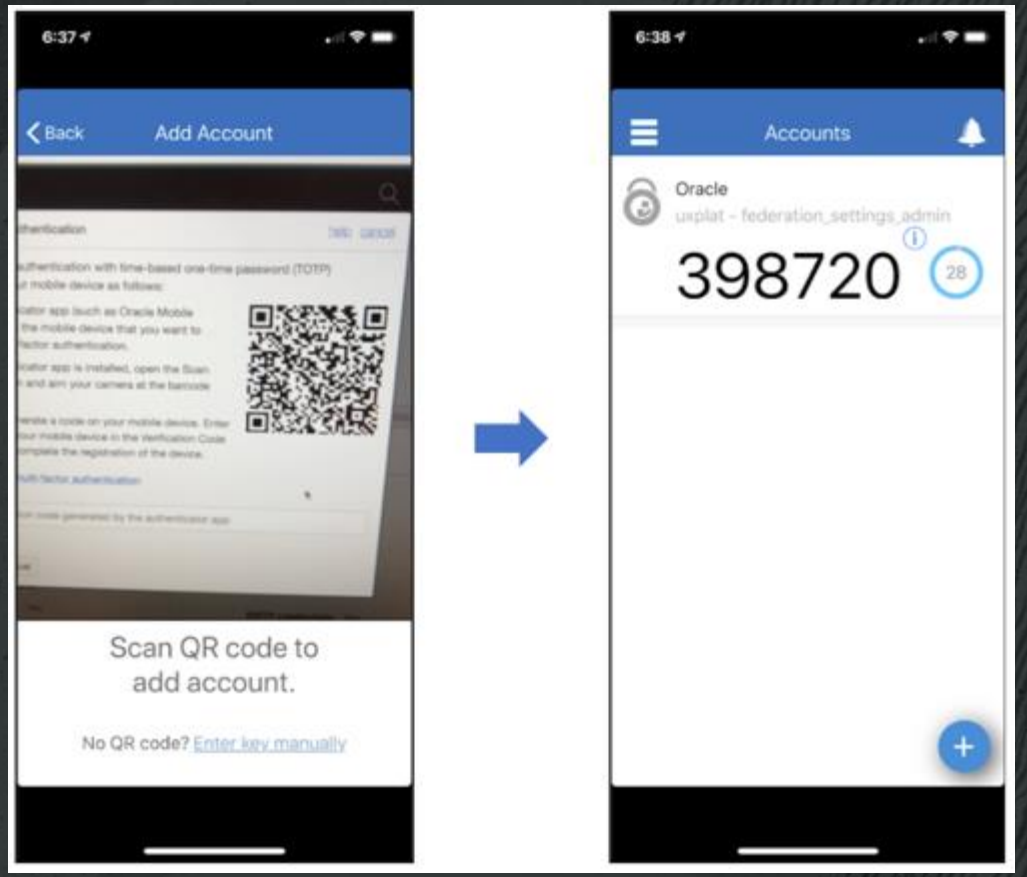
Multi-factor Authentication (MFA) - How to enable?

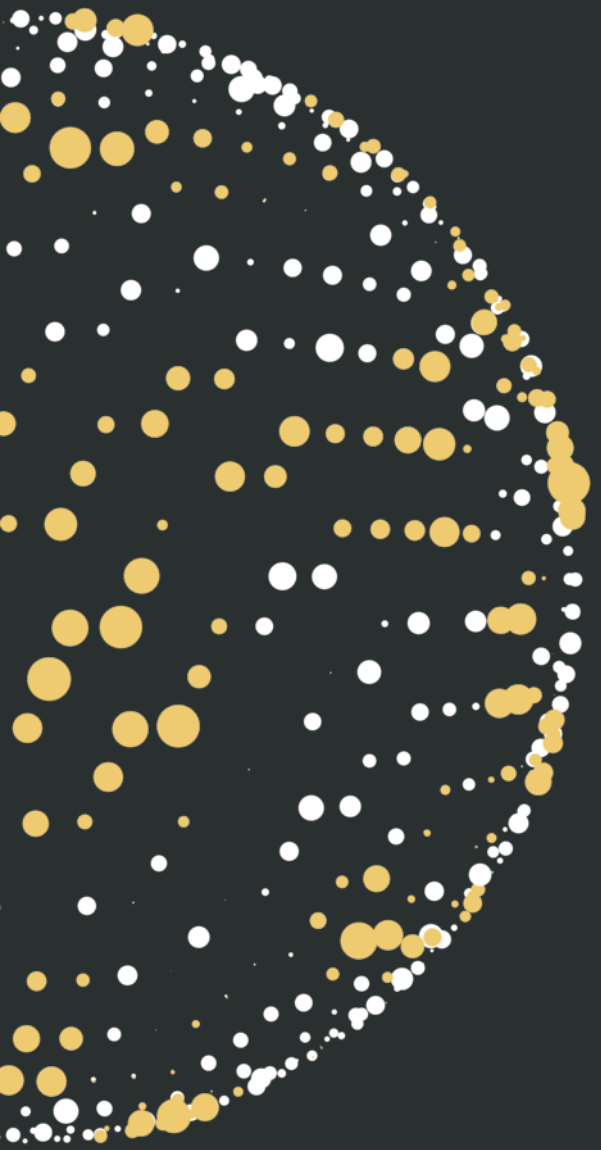
Step 1



Step 2

Step 3





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 <action> <resource-type> in <placement> 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

Policy Syntax

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

Conditions:
Syntax for a single condition: variable =|!= value

- 2 variable types: request (relevant to the request itself), and 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

request.operation	The API operation name being requested
request.permission	The underlying permission(s) requested
request.user.id	OCID of the requesting user
request.groups.id	The OCIDs of groups requesting user is in
target.compartment.id	The OCID of the compartment
target.compartment.name	The name of the compartment specified in target.compartment.id
request.region	The key of the region the request is made in
request.ad	The name of the AD the request is made in

Example: `Allow group Phoenix-Admins to manage all-resources in tenancy where request.region='phx'`

Policy Syntax

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

- Conditions: Syntax for a single condition: variable =|!= value

Type	Types of value
String	(single quotation marks are required around the value)
Pattern	/HR*/ (matches strings that start with "HR") /*HR/ (matches strings that end with "HR") /*HR*/ (matches strings with "HR")

- Syntax for multiple conditions: any|all {<condition>,<condition>,...}

```
Allow group XYZ to manage groups in tenancy
where any {request.operation='ListGroups',
           request.operation='GetGroup',
           request.operation='CreateGroup',
           request.operation='UpdateGroup'}
```

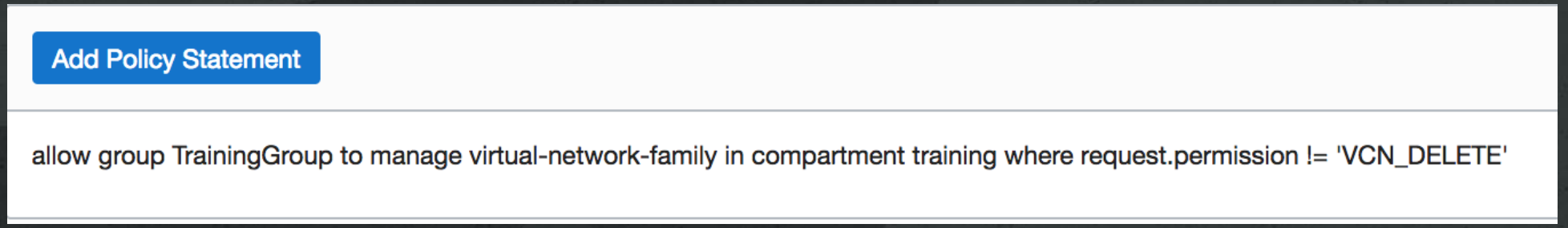
```
Allow group XYZ to manage groups in tenancy
where all {request.permission='GROUP_INSPECT',
           request.operation='ListGroups'}
```

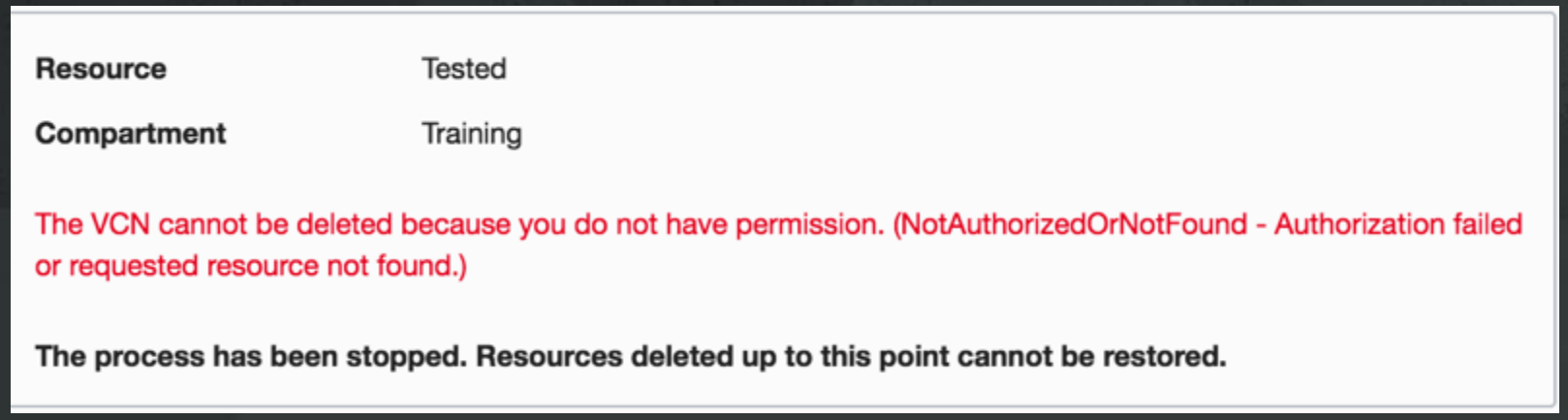
Advanced Policy

- Policy for GroupAdmins group to manage any groups with names that start with "A-Users-"
  - `Allow group GroupAdmins to manage groups in tenancy where target.group.name = /A-Users-*/`
- Policy for GroupAdmins group to manage the membership of any group besides the Administrators group:
  - `Allow group GroupAdmins to use users in tenancy where target.group.name != 'Administrators'`
- Policy lets A-Admins create, update, or delete any groups whose names start with "A-", except for the A-Admins group itself
  - `Allow group GroupAdmins to manage groups in tenancy where all {target.group.name=/A-*/,target.group.name!='A-Admins'}`

Scoping Access with Permissions or API Operations

- In a policy statement, you can use conditions combined with permissions or API operations to reduce the scope of access granted by a particular verb.
- Allow a user to manage VCN resources except have the ability to delete a VCN
  - allow group TrainingGroup to manage virtual-network-family in compartment training where request.permission != 'VCN_DELETE'





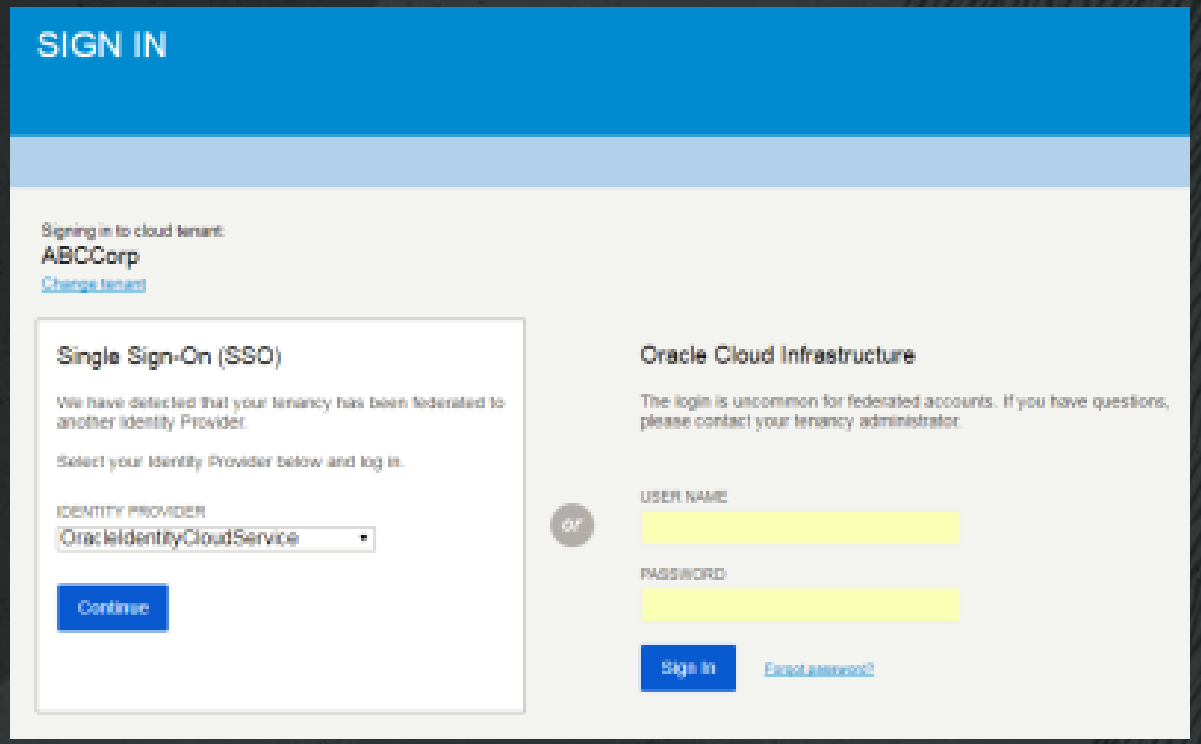
Identity Federation

Best Practices for securing IAM – IAM Federation

- Oracle recommends that you use federation to manage logins into the Console
- An administrator needs to set up a federation trust between the on-premises identity provider (IdP) and IAM, in addition to creating mapping between on-premises groups and IAM groups
- Federation is especially important for enterprises using custom policies for user authentication (for example, multifactor authentication).
- When using federation, Oracle recommends that you create a federation administrators group that maps to the federated IdP administrator group
- The federation administrators group will have administrative privileges to manage customer tenancy, while being governed by the same security policies as the federated IdP administrator group
- In this scenario, it is a good idea to have access to the local tenancy administrator user (that is, member of the default tenancy administrator IAM group), to handle any break-glass type scenarios (for example, inability to access resources through federation)
- Tenancies federated with Oracle Identity Cloud Service or Okta, can leverage SCIM (System for Cross-domain Identity Management). Such users can have the additional credentials such as API keys and auth tokens that are managed in the User Settings page.

Experience for Federated Users

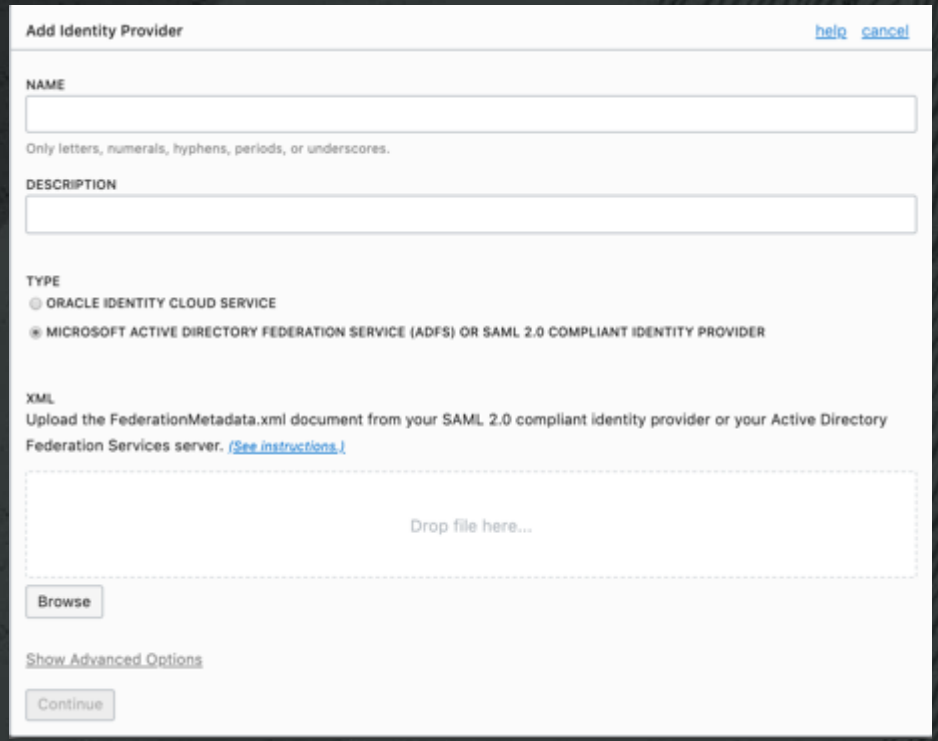
- Federated users can use the Console to access Oracle Cloud Infrastructure (according to IAM policies for the groups the users are in).
- They'll be prompted to enter their Oracle Cloud Infrastructure tenant (for example, ABCCorp).
- They then see a page with two sets of sign-in instructions: one for federated users and one for non-federated (Oracle Cloud Infrastructure) users.



Federation with IDCS - Recommended

Federation with Microsoft Active Directory

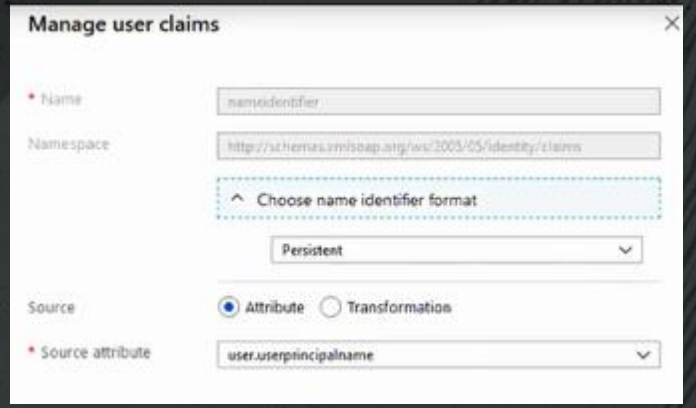
- Get required information from Active Directory Federation Services.
- Federate Active Directory with Oracle Cloud Infrastructure:
  - Add the identity provider (AD FS) to your tenancy and provide the required information.
  - Map Active Directory groups to IAM groups.
- In Active Directory Federation Services, add Oracle Cloud Infrastructure as a trusted, relying party.
- In Active Directory Federation Services, add the claim rules required in the authentication response by Oracle Cloud Infrastructure.
- Test your configuration by logging in to Oracle Cloud Infrastructure with your Active Directory credentials.

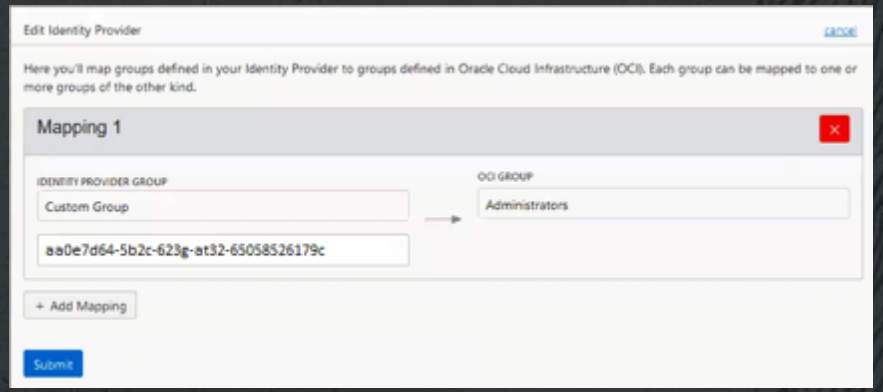


https://docs.cloud.oracle.com/iaas/Content/Identity/Tasks/federatingADFS.htm

Federation with Microsoft Azure Active Directory

- In Oracle Cloud Infrastructure, download the federation metadata document.
- In Azure AD, set up Oracle Cloud Infrastructure Console as an enterprise application.
- In Azure AD, configure the Oracle Cloud Infrastructure enterprise application for single sign-on.
- In Azure AD, set up the user attributes and claims.
- In Azure AD, assign user groups to the application.
- In Azure AD, download the Azure AD SAML metadata document.
- In Oracle Cloud Infrastructure, set up Azure AD as an identity provider.
- In Oracle Cloud Infrastructure, map your Azure AD groups to Oracle Cloud Infrastructure groups.
- In Oracle Cloud Infrastructure, set up the IAM policies to govern access for your Azure AD groups.
- Share the Oracle Cloud Infrastructure sign-in URL with your users.





https://docs.cloud.oracle.com/iaas/Content/Identity/Tasks/federatingADFSazure.htm

Reference IAM Model: Authentication and user management

All access by humans go through federation with a customer's corporate identity provider (IdP) to leverage their proven Auth mechanisms (MFA) and management capabilities (password complexity/rotation)

Use case	Feature
Human using console	Use SAML2.0 federation between corporate IdP and OCI IAM
Human using the CLI/SDK	Create a federated user with SCIM or an OCI IAM user with an API signing key
Human using a PaaS/SaaS app	Use SAML2.0 federation between corporate IdP and OCI IAM
Code running in OCI that calls OCI native APIs	Use Instance Principals
Code running outside OCI that calls OCI APIs	Create an OCI IAM "user" with an API signing key. The "user" in this case represents a software agent, not a human

"Break-glass" access by a human when federation fails	• Create an OCI IAM user in the default Admins group
	• Set a random Console password of sufficient length/complexity
	  • Store this password in a software password manager or physical safe
	  • Password is for infrequent use and should not be human memorizable
	  • Use once – rotate password after every use
	• Monitor via Audit Service
	  • Alarm on any use or attempted use of "break-glass" user
	• Outside the "break-glass" scenario, there is no reason to have an OCI IAM user with a Console password

Reference IAM model for Enterprises

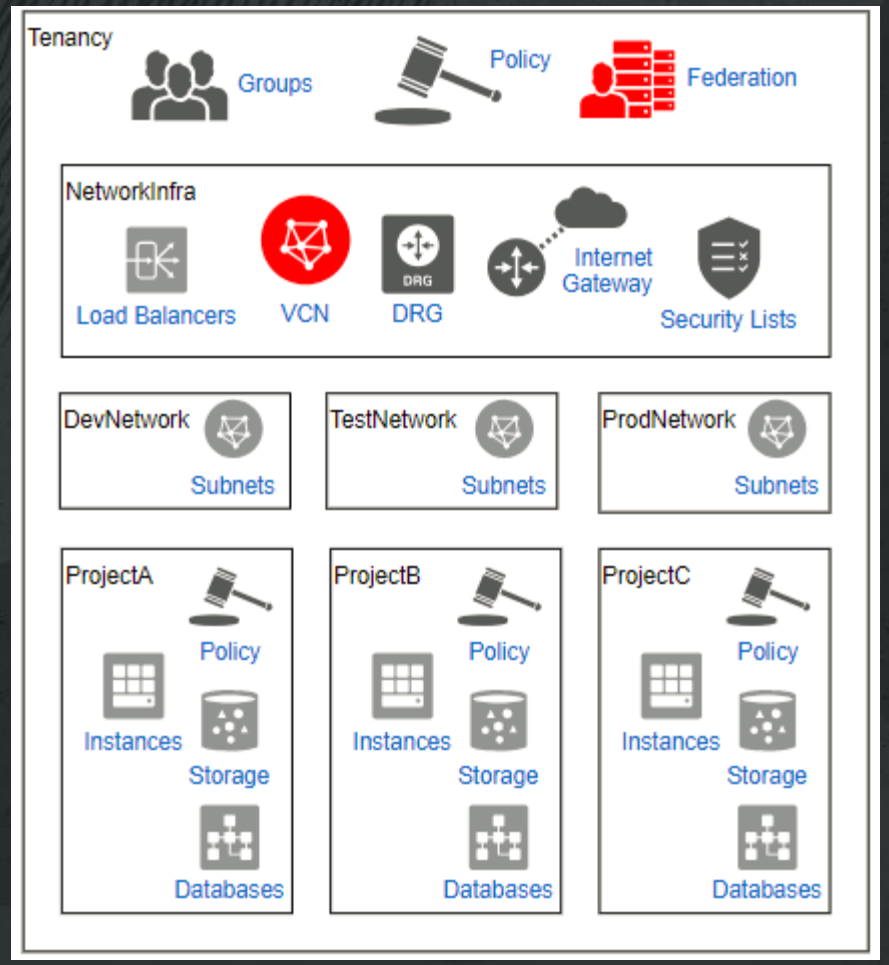
Compartments

- 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:
  - When creating a resource (for example, instance, block storage volume, VCN, subnet), you must decide in which compartment to put it.
  - Compartments are logical, not physical, so related resource components can be placed in different compartments.
  - You can create a hierarchy of compartments up to six compartments deep under the tenancy (root compartment).
  - When you write a policy rule to grant a group of users access to a resource, you always specify the compartment to apply the access rule to. So if you choose to distribute resources across compartments, remember that you will need to provide the appropriate permissions for each compartment for users that will need access to those resources.
  - If you want to delete a compartment, you must delete all resources in the compartment first.

Moving a Compartment to a Different Parent Compartment

- Be aware of following complications before you move a compartment:
    - Required IAM Policy
    - Understanding the Policy Implications When You Move a Compartment
    - Understanding Compartment Quota Implications When You Move a Compartment
    - Understanding Tagging Implications When You Move a Compartment

Reference IAM Model: Compartments

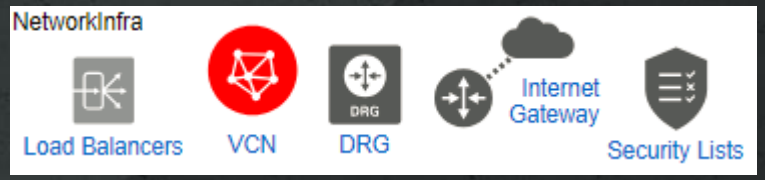


- Compartment: NetworkInfra
  - Critical network infrastructure that should be centrally managed by network admins
  - Resources: Security Lists, Internet Gateways, DRGs, the top-level VCN(s), etc.
- Compartment: ProdNetwork
  - Production environment that may or may not be centrally managed but is typically under change management
  - Modeled as a separate compartment to easily write policy about who can use (i.e. attach resources to) the network
  - Optionally Databases and Storage may be included here depending on whether they are shared resources or not
  - Resources: Subnets, (Databases), (File Storage)

Reference IAM Model: Compartments

Tenancy

Groups
NetworkAdmins (John)



- Allow group NetworkAdmins to MANAGE virtual-network-family in compartment NetworkInfra
- Allow group NetworkAdmins to manage instance-family in compartment NetworkInfra

- John creates a Network in NetworkInfra compartment
- John can't terminate, reboot or launch new instances into ProjectA compartment

Groups
A-Admins (Tom)

ProjectA

- Allow group A-Admins to USE virtual-network-family in compartment NetworkInfra
- Allow group A-Admins to manage all-resources in compartment ProjectA

- Tom launches instances in ProjectA using the VCN in NetworkInfra compartment
- Tom cannot launch instance inside the NetworkInfra compartment

The instances Tom launched reside in the VCN from a network topology standpoint but from an access standpoint, they're in the ProjectA compartment, not the NetworkInfra compartment where the VCN is

Wrap-Up
You should now be familiar with:

- Advanced Policy Syntax
- Multi-factor Authentication
- Federating OCI with Oracle Identity Cloud Service (IDCS)
- Federating OCI with Microsoft Active Directory
- Federating OCI with Microsoft Azure Active Directory
- Reference IAM model
- Compartments design
- Example IAM compartment and policy design

Feedback/Comments/Ask for help

alexandre.af.fagundes@oracle.com

Thank you