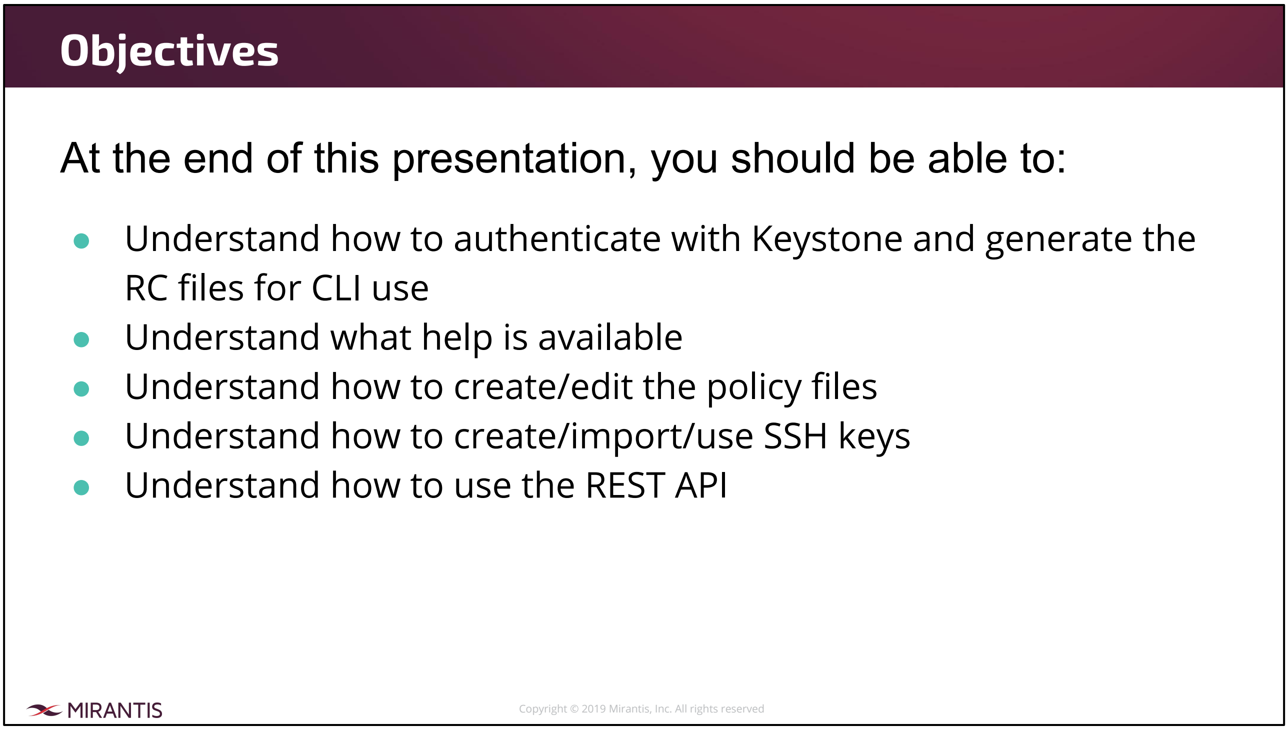
MIRANTIS

Module 2.5: Miscellaneous Topics

Additional topics not covered Day 1

training.mirantis.com

If you are taking the ***Certified OpenStack Administrator (COA)*** exam, the topics discussed in this lecture should be treated as pre-requisites for the exam.
You need to understand each topic (except for the REST API slides) to be successful in the exam.



Objectives

At the end of this presentation, you should be able to:

- Understand how to authenticate with Keystone and generate the RC files for CLI use
- Understand what help is available
- Understand how to create/edit the policy files
- Understand how to create/import/use SSH keys
- Understand how to use the REST API

During the OCM100 exam, you will be required to:

- Generate and use RC files
- Edit policy files
- Create/import/use SSH keys
- Use the CLI and Dashboard UI

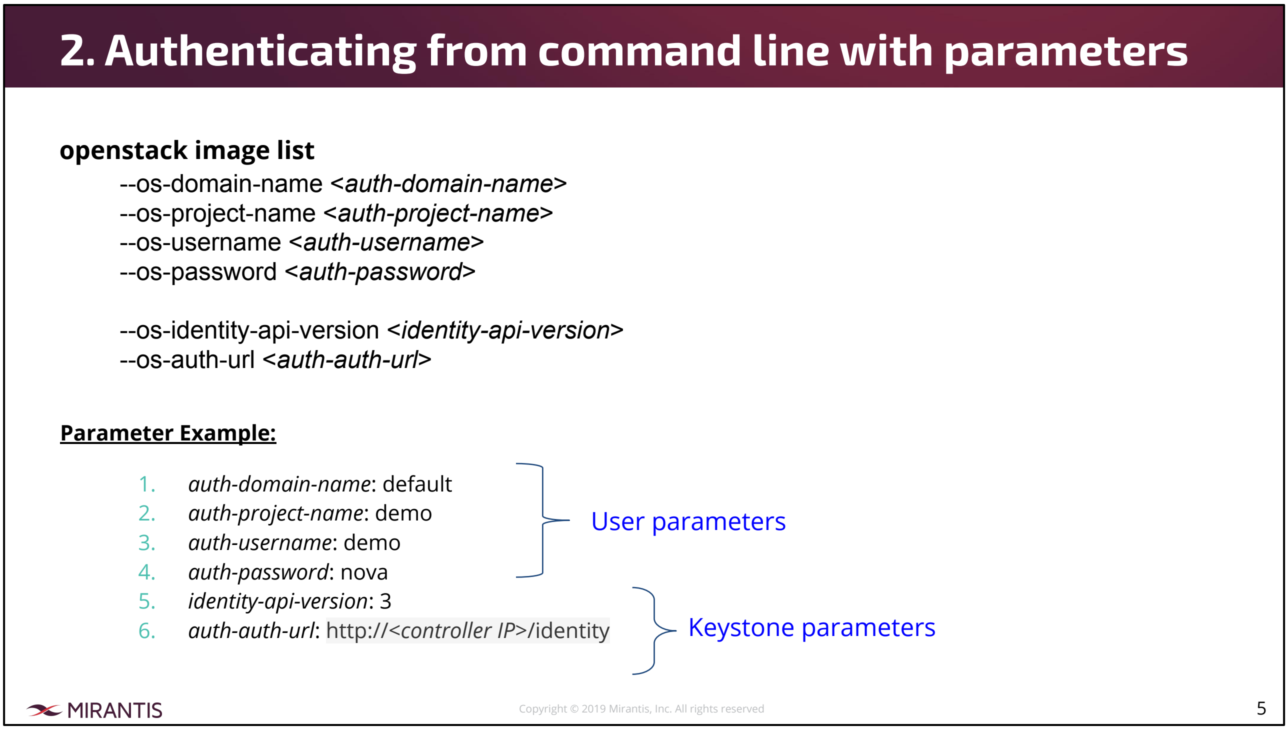
Authenticating with Keystone

Requests (deploy VM, create user, etc.)

require authentication

3 ways to authenticate with Keystone

1. Use the Dashboard UI
    a. Domain **+** user name **+** password (**+** project)
2. Specify additional parameters on CLI command
    a. openstack image list …
3. Specify information in ENV variables
    a. Defined in a “RC” file
    b. *Source* the “RC” file
    c. No need to specify with CLI command



2. Authenticating from command line with parameters

openstack image list

```
--os-domain-name <auth-domain-name>
--os-project-name <auth-project-name>
--os-username <auth-username>
--os-password <auth-password>

--os-identity-api-version <identity-api-version>
--os-auth-url <auth-auth-url>
```

Parameter Example:

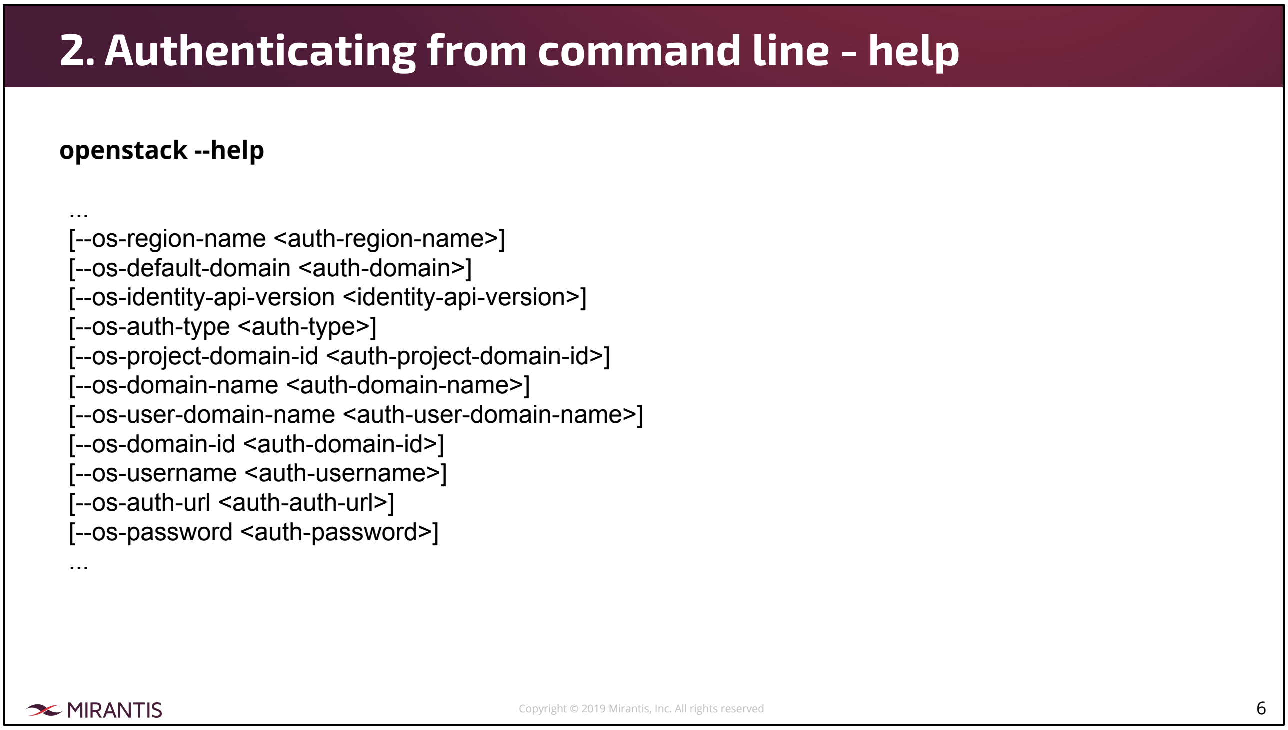
1. *auth-domain-name*: default
2. *auth-project-name*: demo
3. *auth-username*: demo
4. *auth-password*: nova
5. *identity-api-version*: 3
6. *auth-auth-url*: http://<*controller IP*>/identity

User parameters

Keystone parameters

MIRANTIS

This slide shows the minimal parameters, required from the command line, for authentication. Most are related to the user, similar to the login from the Dashboard UI. 2 parameters are needed to define Keystone.



2. Authenticating from command line - help

openstack --help

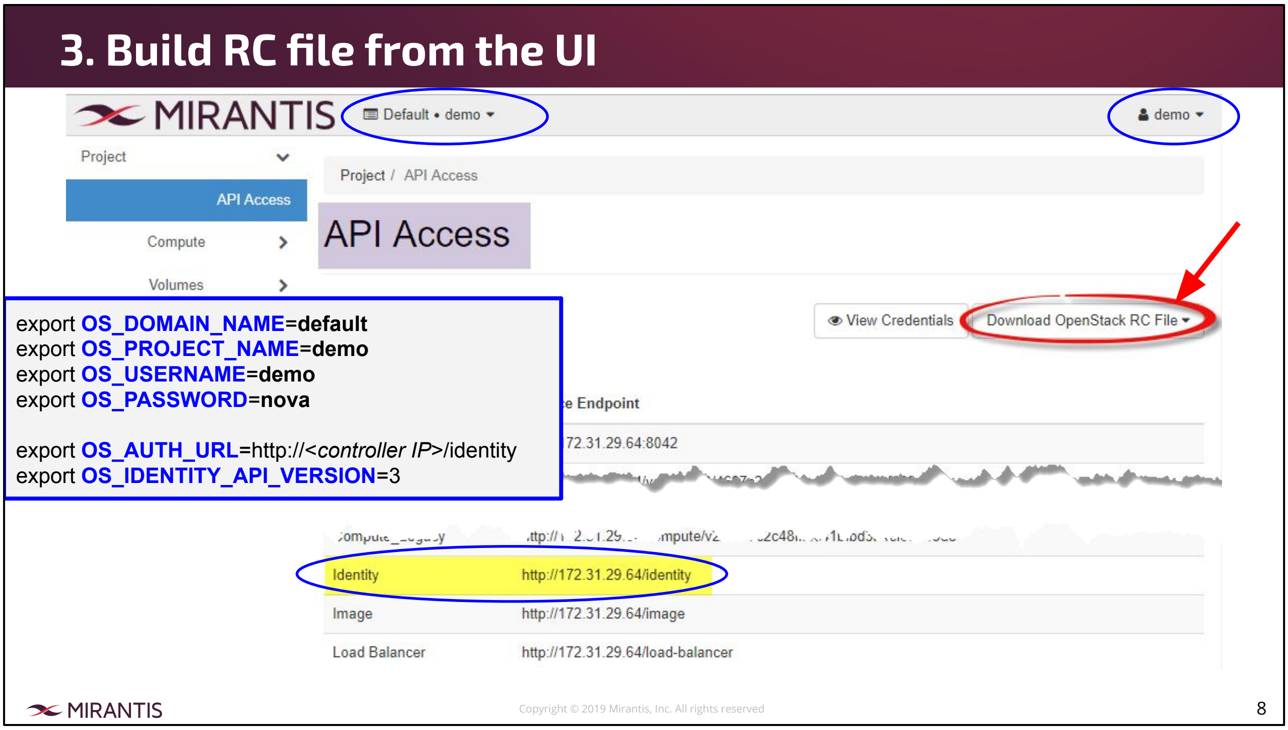
```
...
[--os-region-name <auth-region-name>]
[--os-default-domain <auth-domain>]
[--os-identity-api-version <identity-api-version>]
[--os-auth-type <auth-type>]
[--os-project-domain-id <auth-project-domain-id>]
[--os-domain-name <auth-domain-name>]
[--os-user-domain-name <auth-user-domain-name>]
[--os-domain-id <auth-domain-id>]
[--os-username <auth-username>]
[--os-auth-url <auth-auth-url>]
[--os-password <auth-password>]
...
```

MIRANTIS

Issue an **openstack --help** command to display the parameters shown on the slide, as well as many others.

3. RC file for command line authentication

- ENV variables can be used to *authenticate* the user credentials (domain, name, password) and assign a *token* for the command line request
  - OS_AUTH_URL: defines the Keystone URL
  - OS_USER_DOMAIN_NAME: user domain (ie; default)
  - OS_PROJECT_NAME: project for the user (ie; demo)
  - OS_USER_NAME: name of the user (ie; demo)
  - OS_USER_PASSWORD: password for the user (ie; nova)
  - Plus a few others
- Typically defined in a **RC** file; for example, *adminrc.sh* or *credrc.sh*, and then *sourced* to set the ENV variables
  - Build **RC** file manually or download from Dashboard UI



3. Build RC file from the UI

```
export OS_DOMAIN_NAME=default
export OS_PROJECT_NAME=demo
export OS_USERNAME=demo
export OS_PASSWORD=nova

export OS_AUTH_URL=http://<controller IP>/identity
export OS_IDENTITY_API_VERSION=3
```

	e Endpoint
	72.31.29.64:8042
Identity	http://172.31.29.64/identity
Image	http://172.31.29.64/image
Load Balancer	http://172.31.29.64/load-balancer

To help you with building the RC file, use the Dashboard UI. Shown here is the *API Access* panel.
Click **Download OpenStack RC File** to download a complete RC file. This downloads a RC file to your local machine that must be FTP'd to the lab environment.

As an alternative, you can build the RC file. Think about what you need when you login to the Dashboard:

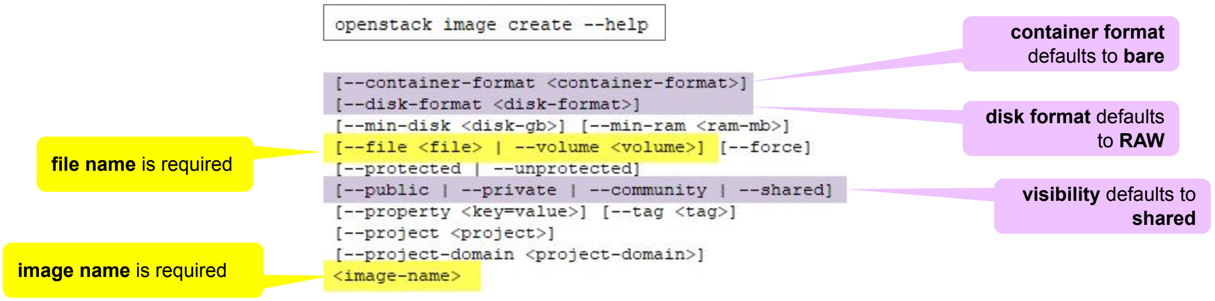
(1) Domain (default)
(2) User name (demo)
(3) Password (nova)
(4) Project (demo)
(5) URL for Keystone (highlighted on the slide)
(6) Version of Keystone API (3)

Access help

MIRANTIS

CLI help

- Save time using the OpenStack (CLI) help:
  - **--help**
  - For example, suppose you need help with uploading (creating) an image:



```
openstack image create --help
```

```
[--container-format <container-format>]
[--disk-format <disk-format>]
[--min-disk <disk-gb>] [--min-ram <ram-mb>]
[--file <file> | --volume <volume>] [--force]
[--protected | --unprotected]
[--public | --private | --community | --shared]
[--property <key=value>] [--tag <tag>]
[--project <project>]
[--project-domain <project-domain>]
<image-name>
```

container format defaults to **bare**

disk format defaults to **RAW**

file name is required

visibility defaults to **shared**

image name is required

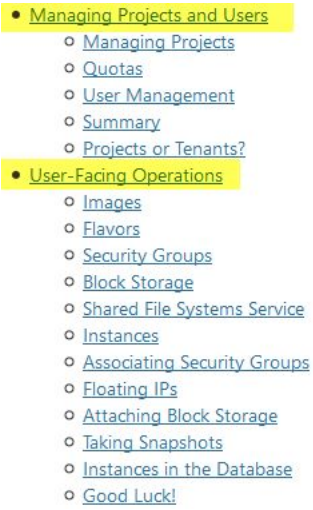
Note: the default values are documented in the remaining help (not shown here)

Help is available during the exam:

- You can use the CLI client to understand the syntax and operands for a command, such as the `openstack image create` command shown on the slide.
  - The help also identifies the default values. For example,
    - disk format = RAW
    - visibility = shared
    - And so on
  - Using the help and the default values, for example, to create an image named *cirrosCOA*, the command might be:
    - `openstack image create --file cirros-0.4.0-x86_64-disk.img cirrosCOA`

OpenStack docs (1)

- In general, you can use the online OpenStack docs to help you
  - docs.openstack.org/rocky/index.html
- The Operations Guide might be helpful:
  - (Example on right)
  - docs.openstack.org/operations-guide/



- Managing Projects and Users
  - Managing Projects
  - Quotas
  - User Management
  - Summary
  - Projects or Tenants?
- User-Facing Operations
  - Images
  - Flavors
  - Security Groups
  - Block Storage
  - Shared File Systems Service
  - Instances
  - Associating Security Groups
  - Floating IPs
  - Attaching Block Storage
  - Taking Snapshots
  - Instances in the Database
  - Good Luck!

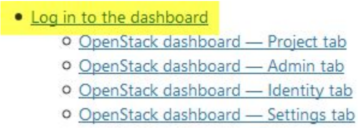
Help is available during the exam:

- You can access docs.openstack.org for help.
- **Tip:** For best results, and to save time, use the Operations Guide.

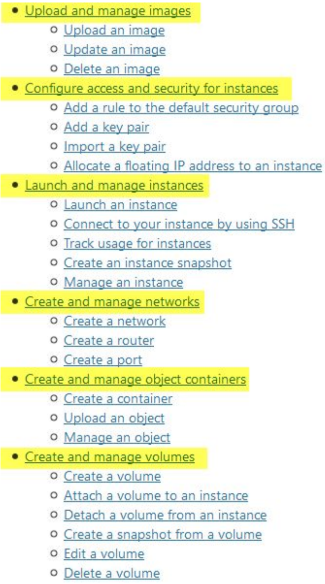
Look carefully at the tasks on the right.

OpenStack docs (2)

- The Dashboard User Guide might be helpful:
  - (Example below and on right)
  - docs.openstack.org/horizon/rocky/user/



- Log in to the dashboard
  - OpenStack dashboard — Project tab
  - OpenStack dashboard — Admin tab
  - OpenStack dashboard — Identity tab
  - OpenStack dashboard — Settings tab

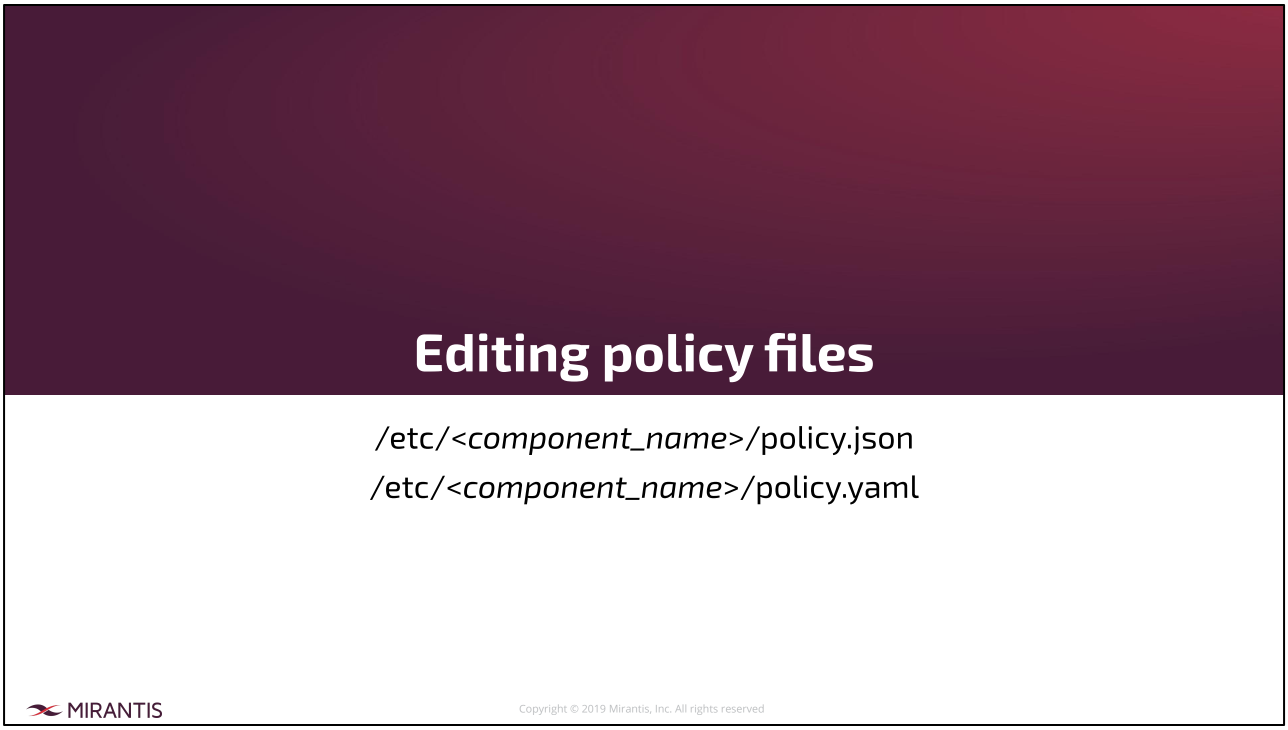


- Upload and manage images
  - Upload an image
  - Update an image
  - Delete an image
- Configure access and security for instances
  - Add a rule to the default security group
  - Add a key pair
  - Import a key pair
  - Allocate a floating IP address to an instance
- Launch and manage instances
  - Launch an instance
  - Connect to your instance by using SSH
  - Track usage for instances
  - Create an instance snapshot
  - Manage an instance
- Create and manage networks
  - Create a network
  - Create a router
  - Create a port
- Create and manage object containers
  - Create a container
  - Upload an object
  - Manage an object
- Create and manage volumes
  - Create a volume
  - Attach a volume to an instance
  - Detach a volume from an instance
  - Create a snapshot from a volume
  - Edit a volume
  - Delete a volume

Help is available during the exam:

- For help with the Dashboard UI, use the Dashboard User Guide.

Look carefully at the tasks on the right.

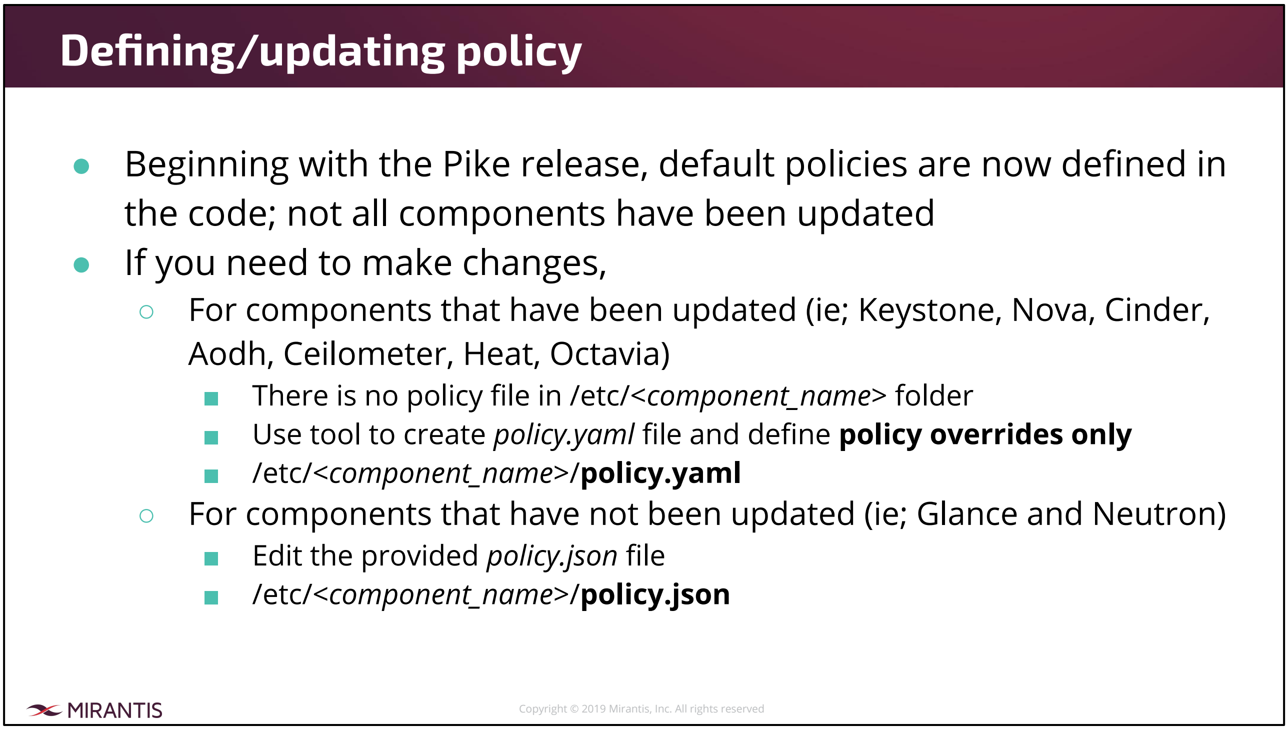


Editing policy files

/etc/*<component_name>*/policy.json

/etc/*<component_name>*/policy.yaml

Each OpenStack service, Identity, Compute, Networking, and so on, has its own role-based access policies. They determine which user can access which objects in which way.
Beginning with the Pike release, the default policies are implemented in the code. You can override the defaults with the **policy.yaml** files.
An ***oslo utility*** is provided to create the **policy.yaml** file first.



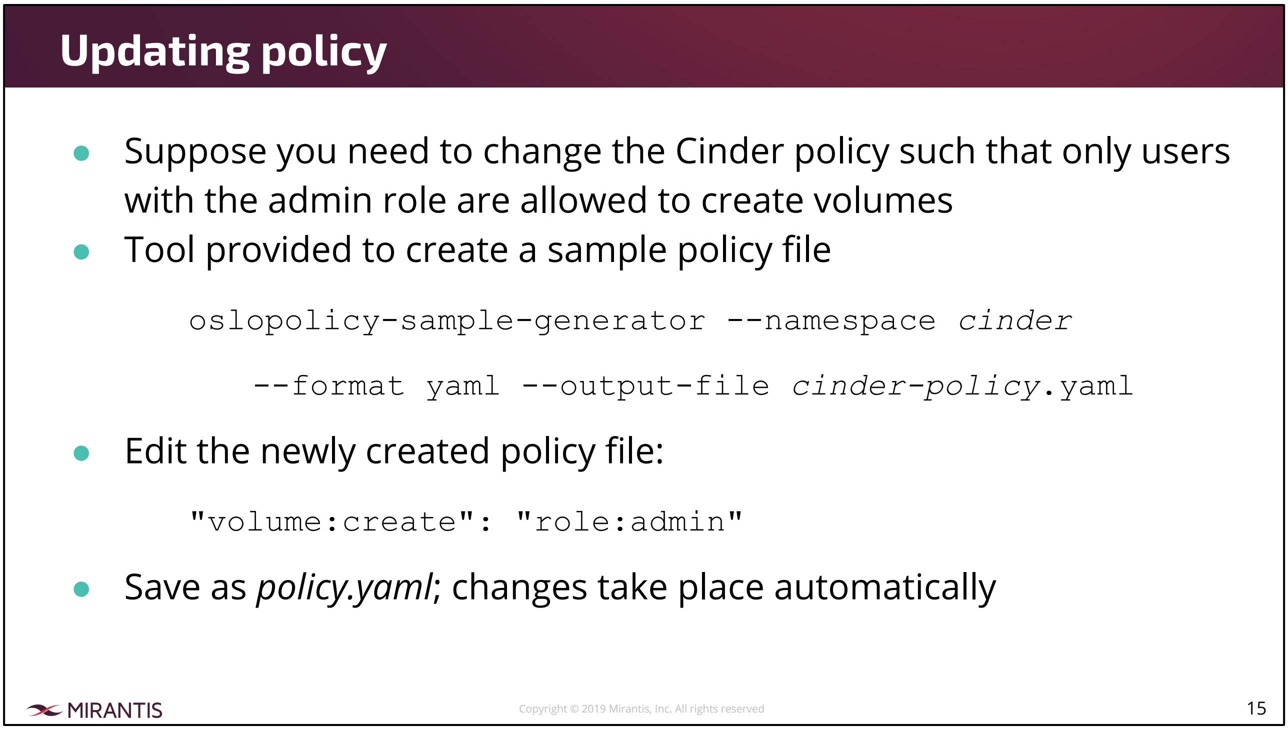
Defining/updating policy

- Beginning with the Pike release, default policies are now defined in the code; not all components have been updated
- If you need to make changes,
  - For components that have been updated (ie; Keystone, Nova, Cinder, Aodh, Ceilometer, Heat, Octavia)
    - There is no policy file in /etc/<*component_name*> folder
    - Use tool to create *policy.yaml* file and define **policy overrides only**
    - /etc/<*component_name*>/**policy.yaml**
  - For components that have not been updated (ie; Glance and Neutron)
    - Edit the provided *policy.json* file
    - /etc/<*component_name*>/**policy.json**

Default policies are defined in the code.

- For components that support the newer technology, simply define policy overrides in its **policy.yaml** file.
- For components that do not support the newer technology, edit/update the **policy.json** file.

More information on the policy.yaml file:
https://docs.openstack.org/ocata/config-reference/policy-yaml-file.html



Updating policy

- Suppose you need to change the Cinder policy such that only users with the admin role are allowed to create volumes
- Tool provided to create a sample policy file

```
oslopolicy-sample-generator --namespace cinder

   --format yaml --output-file cinder-policy.yaml
```

- Edit the newly created policy file:

```
"volume:create": "role:admin"
```

- Save as *policy.yaml*; changes take place automatically

MIRANTIS

If there is no policy file, use the **oslopolicy-sample-generator** tool to build one. All policies and rules will be commented out. The example on this slide shows how to apply the **admin role** to the **create volume** operation. The change is immediate - no need to restart the OpenStack services.

In this case, you edit the yaml file:

(1) Locate the create volume policy
(2) Uncomment it (all policy rules are commented out)
(3) Modify it

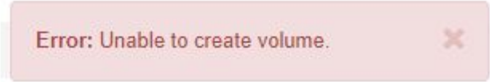
More details on the policy generator tool:
https://docs.openstack.org/oslo.policy/latest/cli/index.html
For more details on configuring the policy files:
https://docs.openstack.org/keystone/rocky/configuration/policy.html

Policy is an area being updated by the OpenStack community. As a result, this is a transition period. For example, Cinder policy configuration uses an older syntax. Other components, such as Nova, use a different (newer) syntax.

Caution: care should be used when updating the policy file(s). If not done properly, you might cause problems with the system. Always backup any files before editing.

Example response - non-admin users

- From UI, non-admin users see



Error: Unable to create volume.

- From CLI, non-admin users see

```
openstack volume create --size 1 nonAdminVOL
Policy doesn't allow volume:create to be performed.  (HTTP 403)
(Request-ID: req-38fa4a71-2221-4107-b4d3-836f0ce95b64)
```

MIRANTIS

SSH keys

Additional SSH session security

MIRANTIS

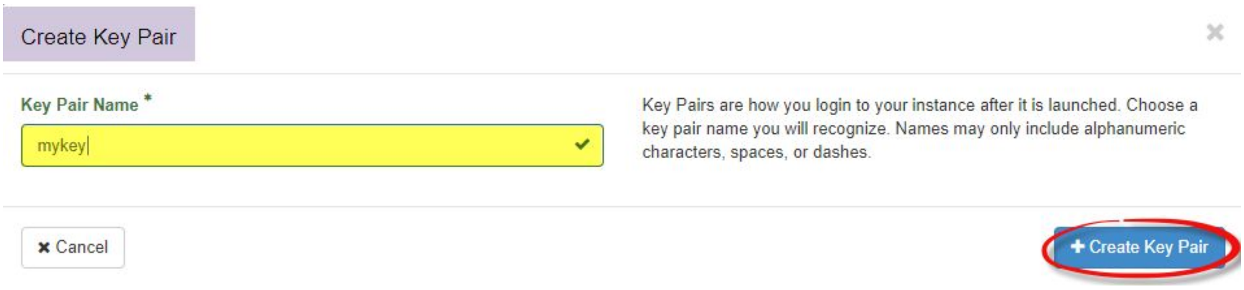
This section discusses use of the Dashboard UI to create/import SSH keys and use the keys to SSH into a deployed VM instance.

SSH keys

- You can define SSH to deny password authentication and instead require a key; giving your instance a much stronger layer of security
- OpenStack can inject an SSH key into instances when deployed
- You use the key for SSH connections, instead of standard name/password credentials

Step 1a. Create SSH key in Dashboard UI

- Project > Compute > Key Pairs : Create Key Pair



Create Key Pair

Key Pair Name *

mykey

Key Pairs are how you login to your instance after it is launched. Choose a key pair name you will recognize. Names may only include alphanumeric characters, spaces, or dashes.

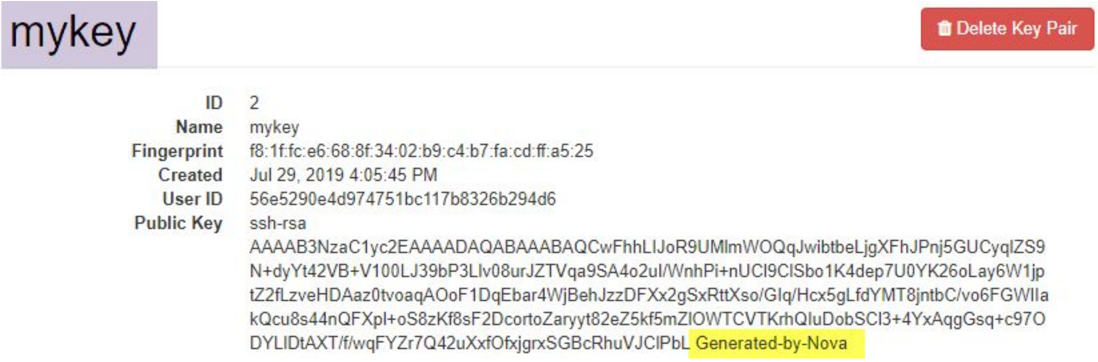
This slide shows how to use the Dashboard UI to create an SSH key.

- Click **Project > Compute > Key Pairs**
- In the Key Pair panel, click **+Create Key Pair**
- This panel is displayed. Define a key pair name, such as, ***mykey*** and click **Create Key Pair**.
- The new SSH key is created and automatically downloaded to your local machine.

You might have SSH keys defined that you wish to use. Use the **Import** function on the Key Pairs panel.

Step 1b. Display SSH key in Dashboard UI

- Project > Compute > Key Pairs : *mykey*



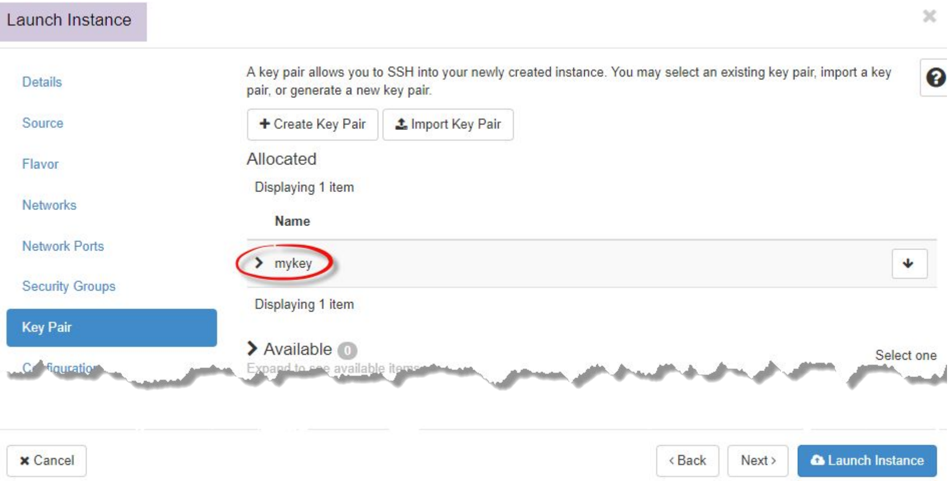
mykey

ID	2
Name	mykey
Fingerprint	f8:1f:fc:e6:68:8f:34:02:b9:c4:b7:fa:cd:ff:a5:25
Created	Jul 29, 2019 4:05:45 PM
User ID	56e5290e4d974751bc117b8326b294d6
Public Key	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCwFhhLIJoR9UMlmWOQqJwibtbeLjgXFhJPnj5GUCyqIZS9 N+dyYt42VB+V100LJ39bP3LIv08urJZTVqa9SA4o2uI/WnhPi+nUCI9CISbo1K4dep7U0YK26oLay6W1jp tZ2fLzveHDAaz0tvoaqAOoF1DqEbar4WjBehJzzDFXx2gSxRttXso/GIq/Hcx5gLfdYMT8jntbC/vo6FGWIIa kQcu8s44nQFXpI+oS8zKf8sF2DcortoZaryyt82eZ5kf5mZIOWTCVTKrhQIuDobSCI3+4YxAqgGsq+c97O DYLIDtAXT/f/wqFYZr7Q42uXxfOfxjgrxSGBcRhuVJCIPbL Generated-by-Nova

This slide shows the contents of the *mykey* SSH key.

Step 2. Deploy instance with SSH key

- Follow usual steps to deploy VM instance
- **Key Pair** tab: select the new SSH key

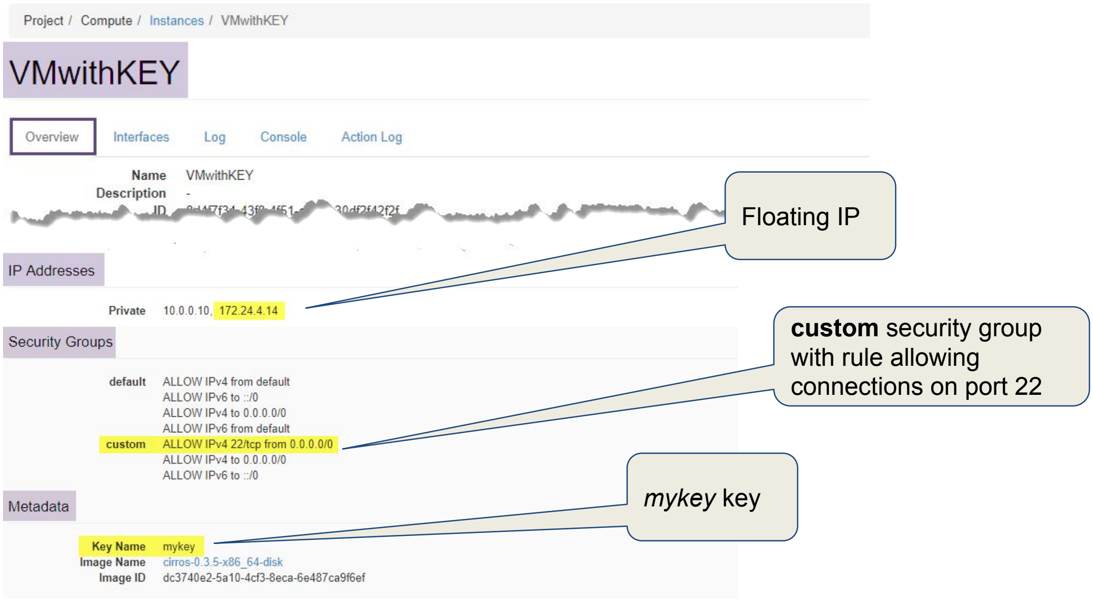


Launch Instance

A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.

At this point, you should understand how to create a VM instance from the Dashboard UI. This slide focuses on the **Key Pair** tab. Select the SSH key on this panel.
Note: You can also import or create a new SSH key on this panel.

Step 2. Instance details



Project / Compute / Instances / VMwithKEY

VMwithKEY

Name	VMwithKEY
Description	-
ID	9d4f7f34-43f9-4f51- 30df2f42f2f

IP Addresses

Private	10.0.0.10, 172.24.4.14

Security Groups

default	ALLOW IPv4 from default ALLOW IPv6 to ::/0 ALLOW IPv4 to 0.0.0.0/0 ALLOW IPv6 from default
custom	ALLOW IPv4 22/tcp from 0.0.0.0/0 ALLOW IPv4 to 0.0.0.0/0 ALLOW IPv6 to ::/0

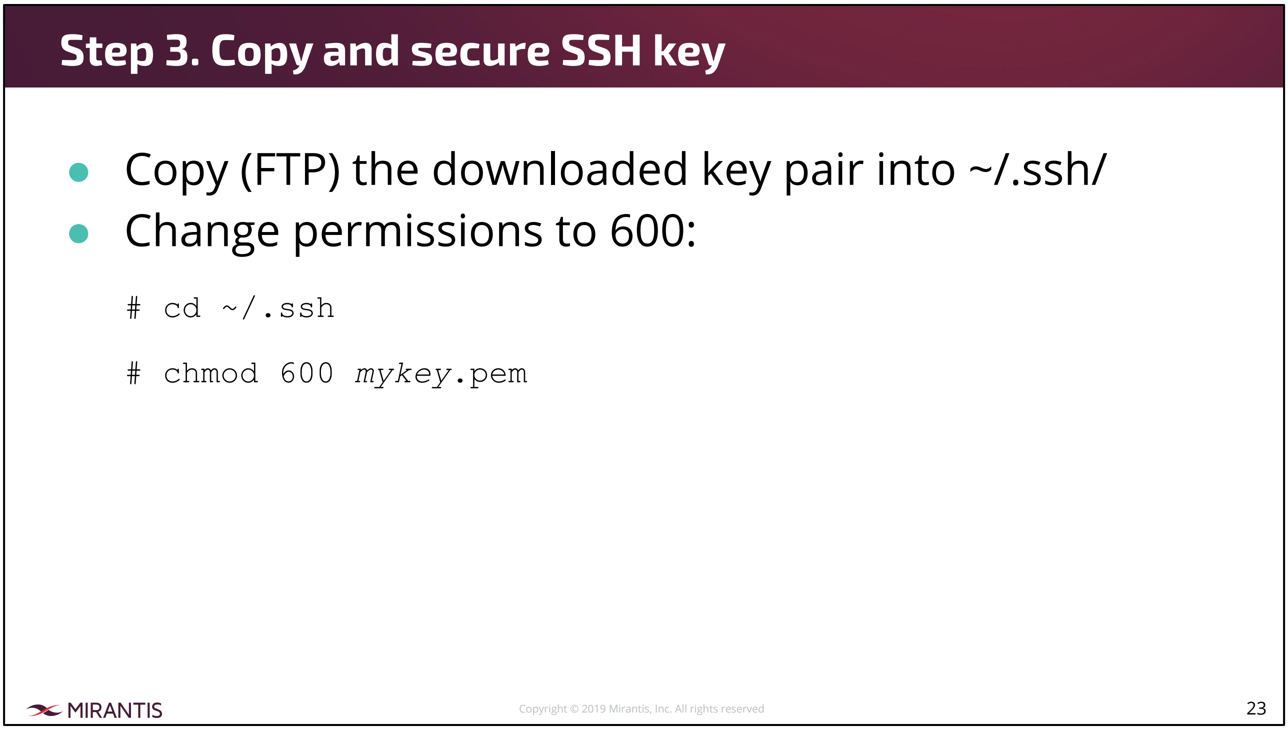
Metadata

Key Name	mykey
Image Name	cirros-0.3.5-x86_64-disk
Image ID	dc3740e2-5a10-4cf3-8eca-6e487ca9f6ef

Floating IP

custom security group with rule allowing connections on port 22

mykey key



Step 3. Copy and secure SSH key

- Copy (FTP) the downloaded key pair into ~/.ssh/
- Change permissions to 600:

```
# cd ~/.ssh

# chmod 600 mykey.pem
```

MIRANTIS

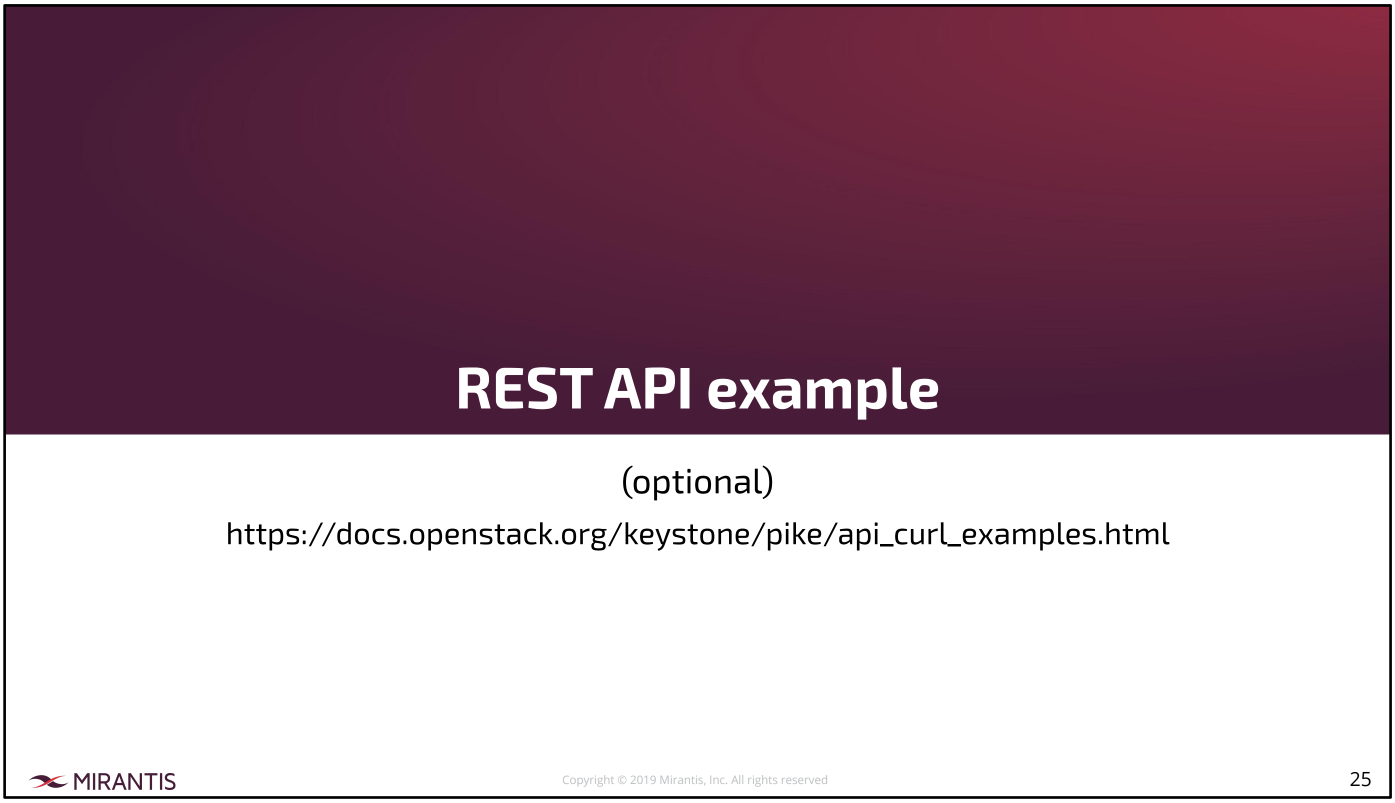
To use your new key pair, you need to make it available to your SSH client. These instructions are for a Linux environment. On Windows, how you use your new key will depend on your client.

Step 4. Connect to instance using SSH key

- Use the key pair to connect to instances that were created using the key pair:

```
# ssh -i ~/.ssh/mykey.pem cirros@floating_IP
```

- **Note:** You still need the following:
  - SSH port open (security group rule)
  - Public IP to access the instance (floating IP address)



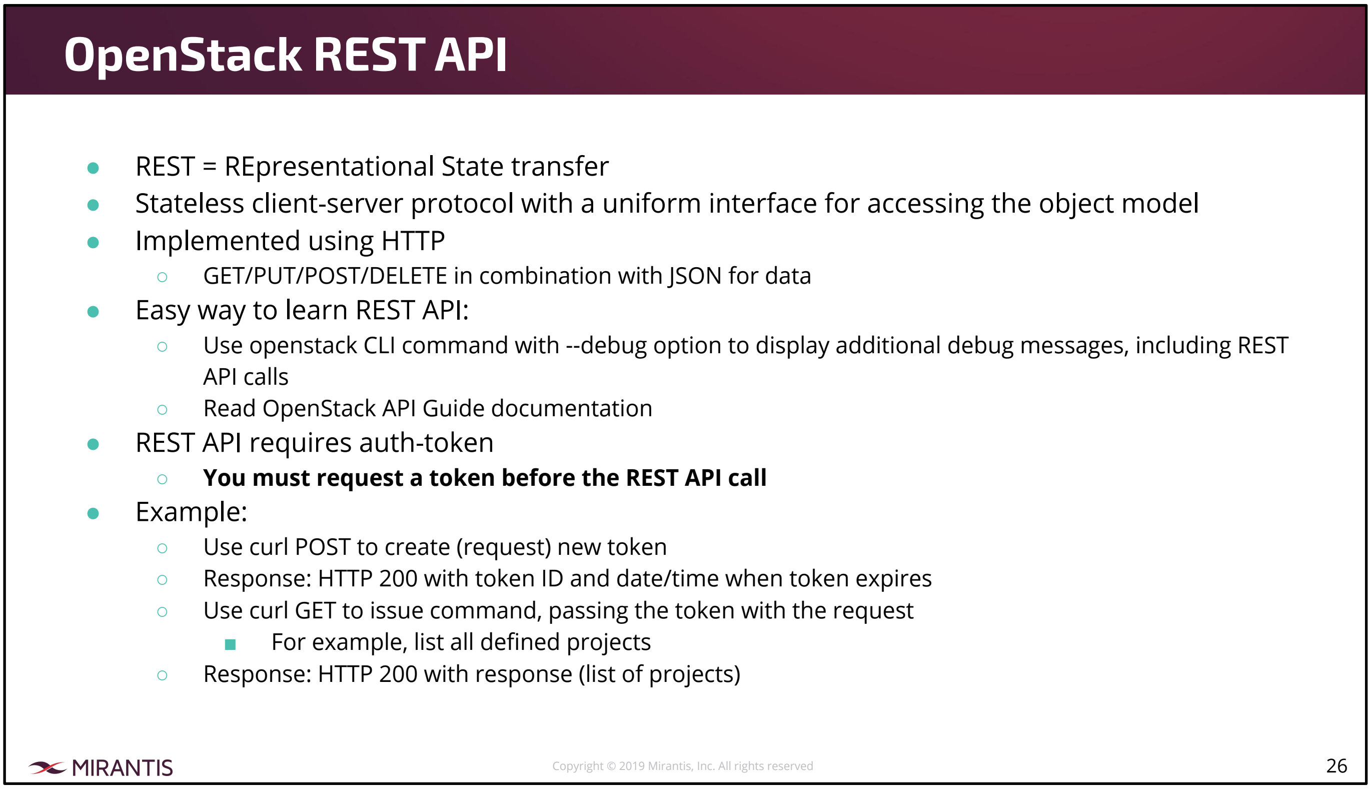
REST API example

(optional)

https://docs.openstack.org/keystone/pike/api_curl_examples.html

MIRANTIS

This lesson discusses how to use the OpenStack REST API – retrieving a valid token and then using that token in a GET request (to display all projects).



OpenStack REST API

- REST = REpresentational State transfer
- Stateless client-server protocol with a uniform interface for accessing the object model
- Implemented using HTTP
  - GET/PUT/POST/DELETE in combination with JSON for data
- Easy way to learn REST API:
  - Use openstack CLI command with --debug option to display additional debug messages, including REST API calls
  - Read OpenStack API Guide documentation
- REST API requires auth-token
  - **You must request a token before the REST API call**
- Example:
  - Use curl POST to create (request) new token
  - Response: HTTP 200 with token ID and date/time when token expires
  - Use curl GET to issue command, passing the token with the request
    - For example, list all defined projects
  - Response: HTTP 200 with response (list of projects)

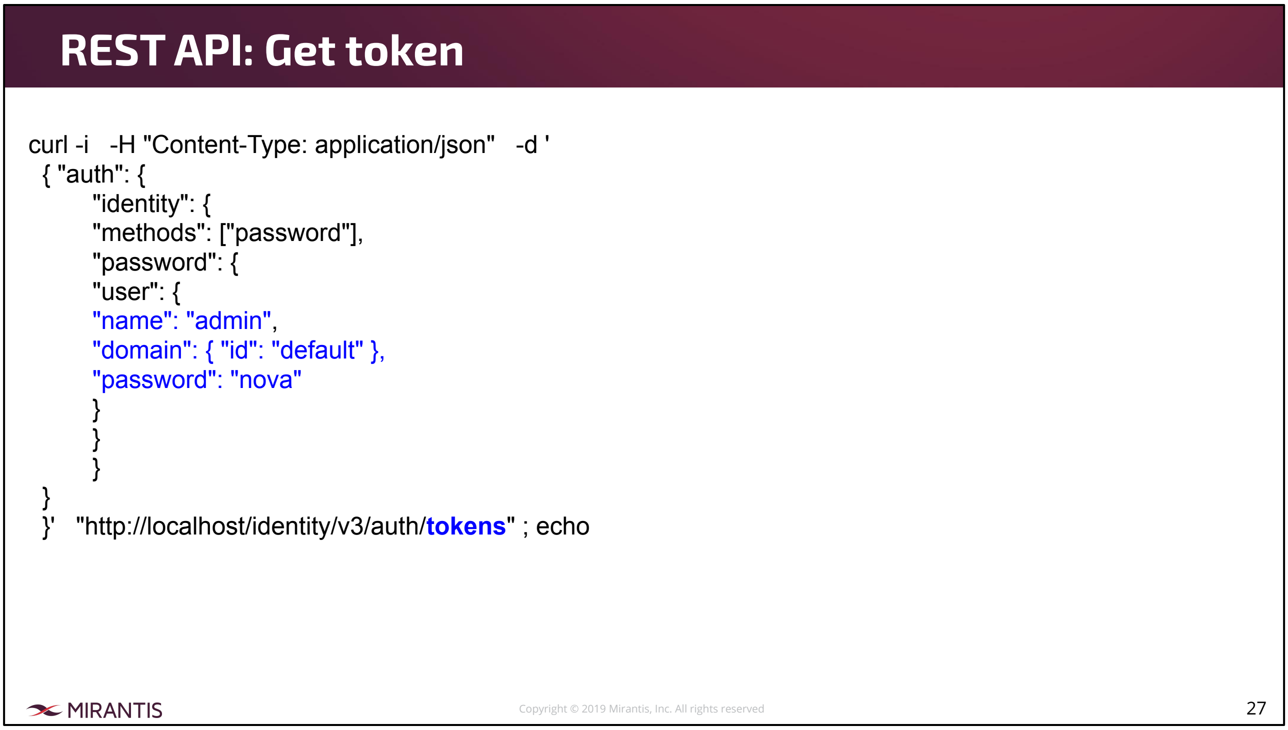
MIRANTIS

OpenStack operations can be performed from the command line interface (CLI), Horizon (Dashboard) UI, or through REST API calls.

Internally, each OpenStack component deals in terms of REST API calls only. Requests from the CLI or Dashboard UI are converted to equivalent REST API calls before being sent to the component. You can see the REST API calls when you use the **--debug** option for the **openstack** command.

For more details on the cURL command:
curl.haxx.se/docs/manpage.html

For more details on the OpenStack APIs, read the *API Guide*:
developer.openstack.org/api-guide/quick-start/

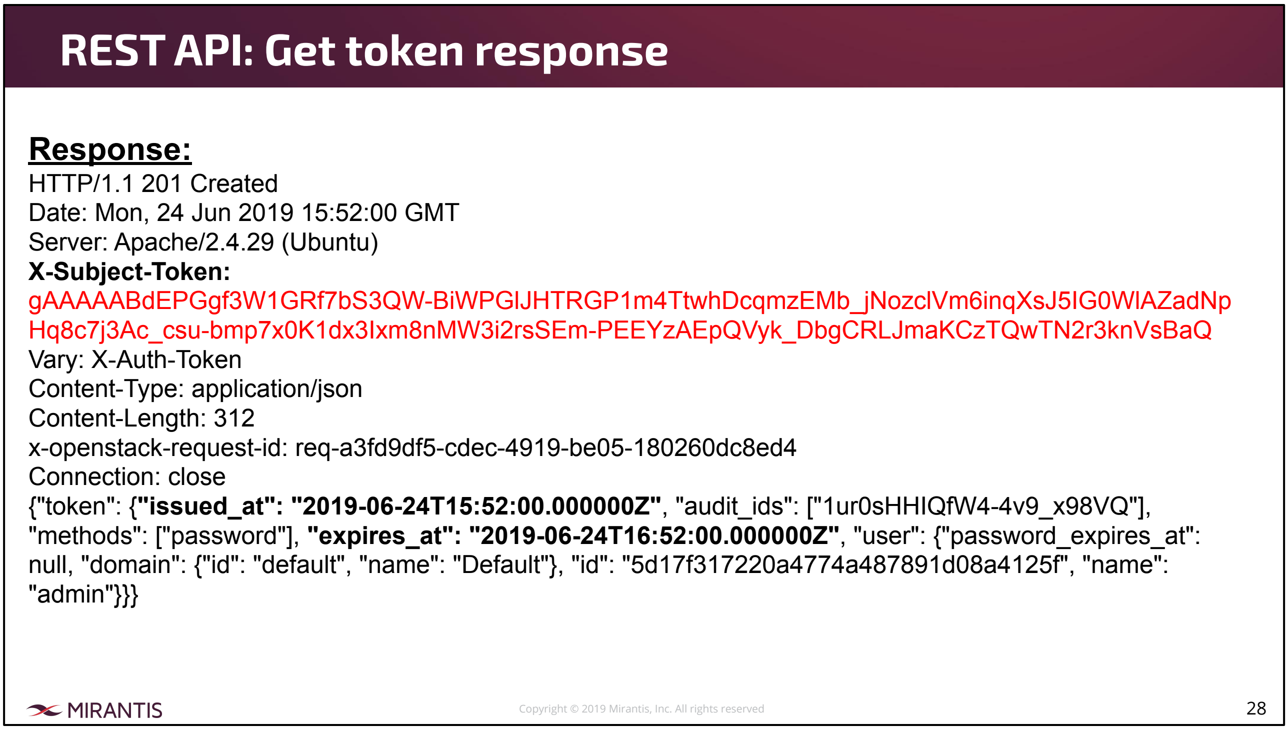


REST API: Get token

```
curl -i   -H "Content-Type: application/json"   -d '
 { "auth": {
        "identity": {
        "methods": ["password"],
        "password": {
        "user": {
        "name": "admin",
        "domain": { "id": "default" },
        "password": "nova"
        }
        }
        }
 }
 }'   "http://localhost/identity/v3/auth/tokens" ; echo
```

This slide shows an example REST API call to get a token. The token must be passed to any REST API request, such as “GET PROJECTS.”

This is the equivalent to an **openstack token issue** command from the CLI.



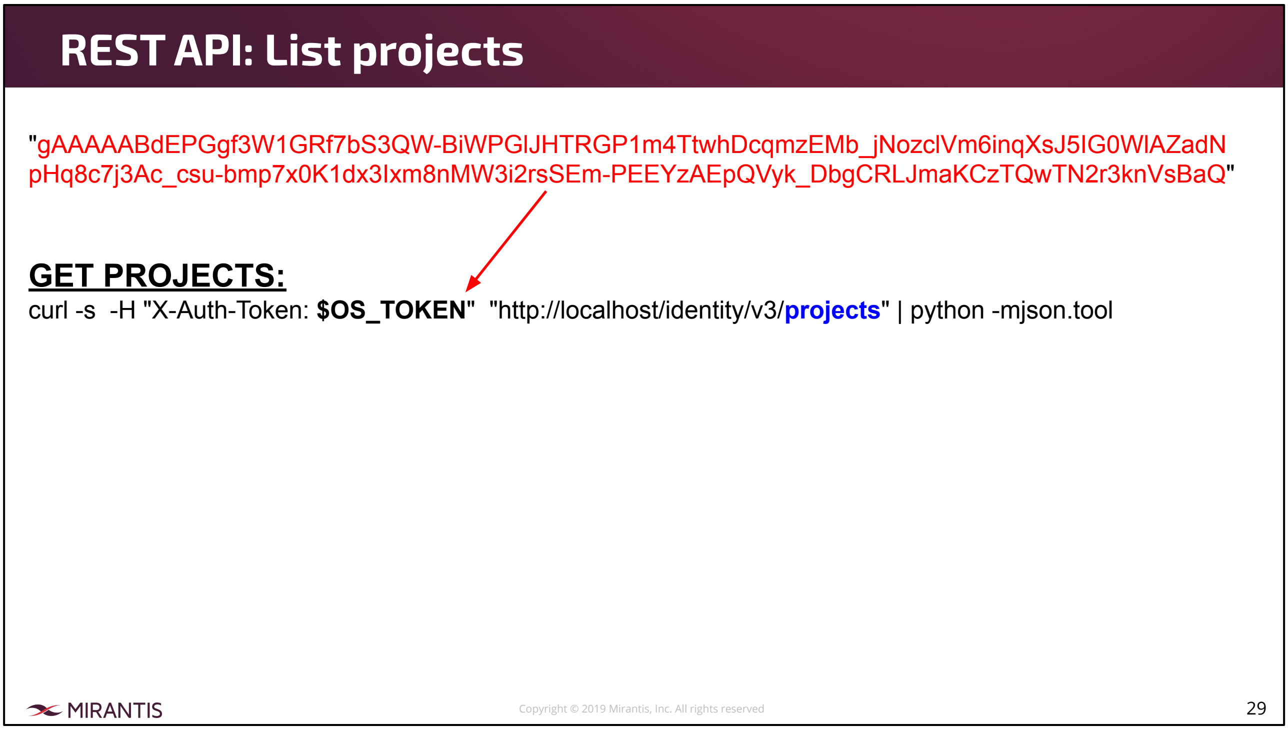
REST API: Get token response

Response:

```
HTTP/1.1 201 Created
Date: Mon, 24 Jun 2019 15:52:00 GMT
Server: Apache/2.4.29 (Ubuntu)
X-Subject-Token:
gAAAAABdEPGgf3W1GRf7bS3QW-BiWPGlJHTRGP1m4TtwhDcqmzEMb_jNozclVm6inqXsJ5IG0WlAZadNp
Hq8c7j3Ac_csu-bmp7x0K1dx3Ixm8nMW3i2rsSEm-PEEYzAEpQVyk_DbgCRLJmaKCzTQwTN2r3knVsBaQ
Vary: X-Auth-Token
Content-Type: application/json
Content-Length: 312
x-openstack-request-id: req-a3fd9df5-cdec-4919-be05-180260dc8ed4
Connection: close
{"token": {"issued_at": "2019-06-24T15:52:00.000000Z", "audit_ids": ["1ur0sHHIQfW4-4v9_x98VQ"],
"methods": ["password"], "expires_at": "2019-06-24T16:52:00.000000Z", "user": {"password_expires_at":
null, "domain": {"id": "default", "name": "Default"}, "id": "5d17f317220a4774a487891d08a4125f", "name":
"admin"}}}
```

MIRANTIS

This slide shows an example REST API response for a “GET TOKEN” request. The token is highlighted in red. By default, the token is valid for 1 hour.



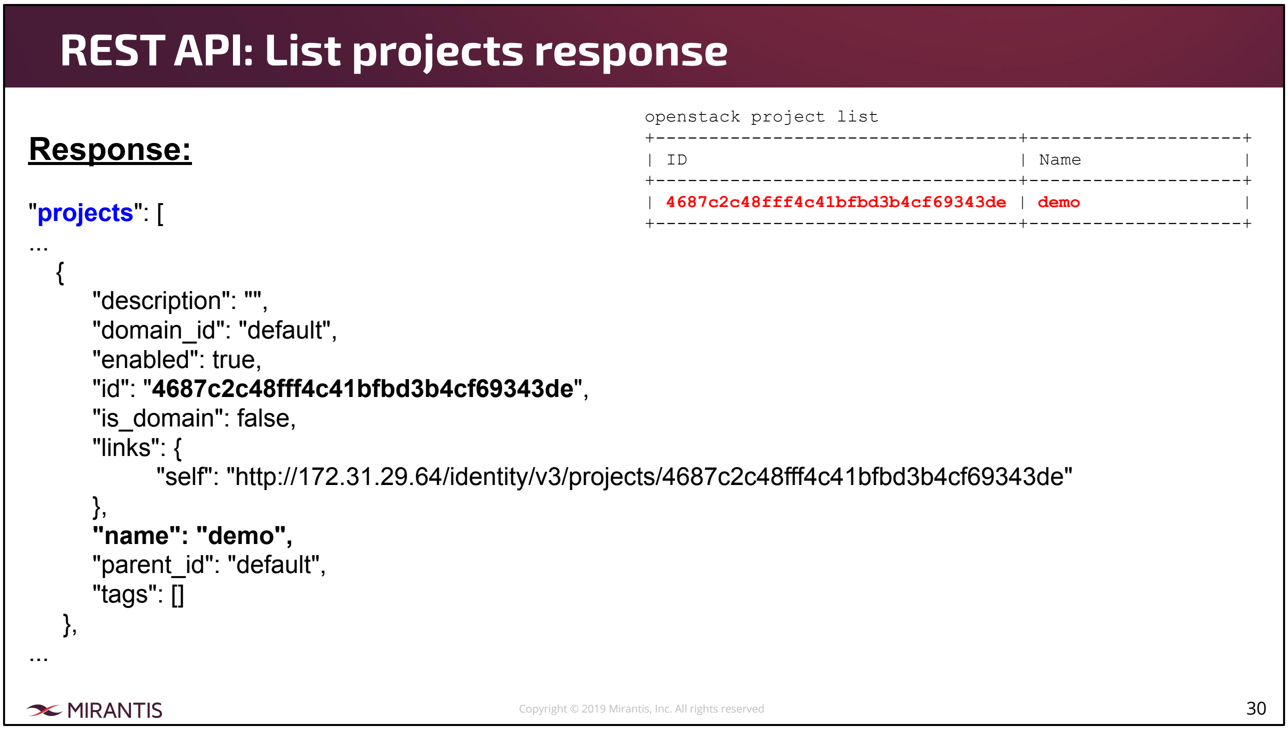
REST API: List projects

```
"gAAAABdEPGgf3W1GRf7bS3QW-BiWPGlJHTRGP1m4TtwhDcqmzEMb_jNozclVm6inqXsJ5IG0WlAZadN
pHq8c7j3Ac_csu-bmp7x0K1dx3Ixm8nMW3i2rsSEm-PEEYzAEpQVyk_DbgCRLJmaKCzTQwTN2r3knVsBaQ"
```

GET PROJECTS:

```
curl -s  -H "X-Auth-Token: $OS_TOKEN"  "http://localhost/identity/v3/projects" | python -mjson.tool
```

Substitute the actual token value in place of the $OS_TOKEN variable in the **curl** command to “GET PROJECTS.”



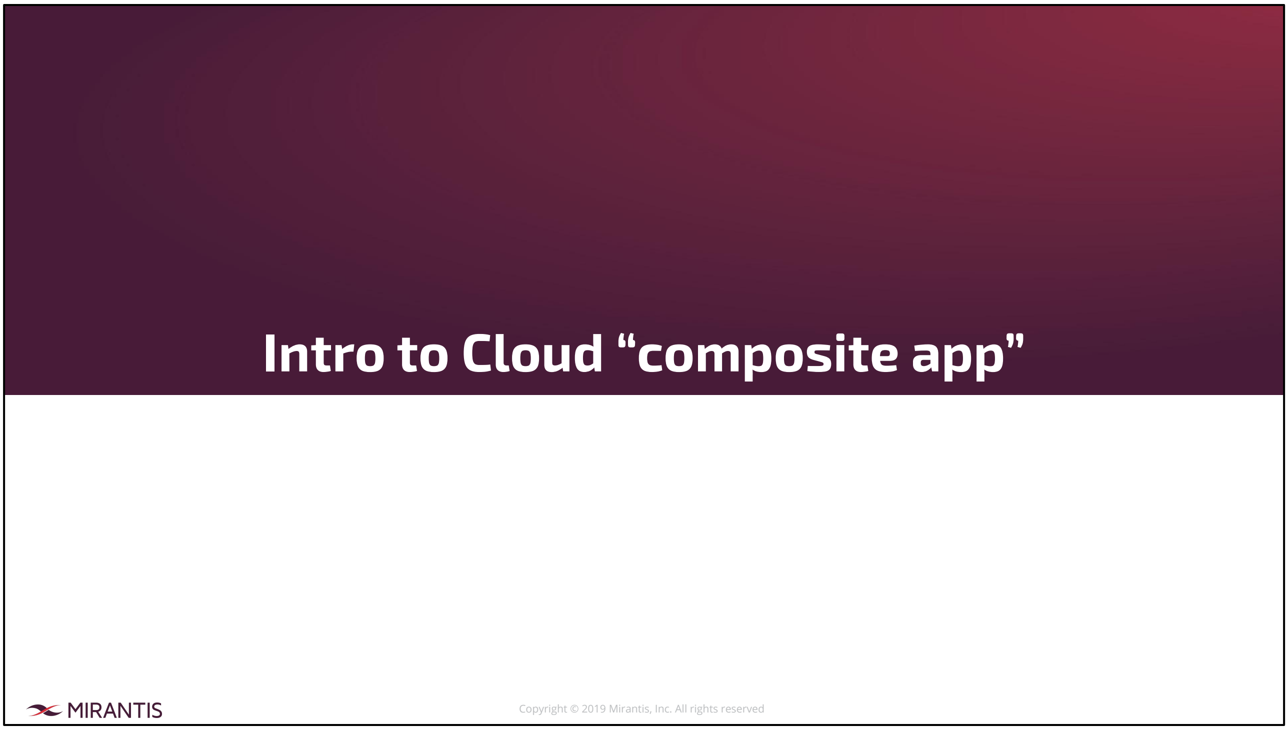
REST API: List projects response

Response:

```
"projects": [
...
   {
       "description": "",
       "domain_id": "default",
       "enabled": true,
       "id": "4687c2c48fff4c41bfbd3b4cf69343de",
       "is_domain": false,
       "links": {
              "self": "http://172.31.29.64/identity/v3/projects/4687c2c48fff4c41bfbd3b4cf69343de"
       },
       "name": "demo",
       "parent_id": "default",
       "tags": []
   },
...
```

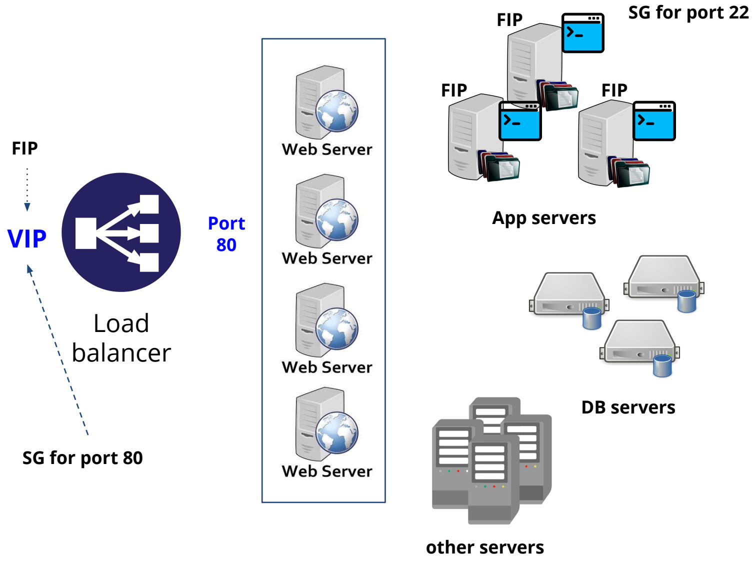
```
openstack project list
+----------------------------------+---------------------+
| ID                               | Name                |
+----------------------------------+---------------------+
| 4687c2c48fff4c41bfbd3b4cf69343de | demo                |
+----------------------------------+---------------------+
```

This slide shows an example REST API response for a “GET PROJECTS” request. In a typical environment, there are multiple projects. This example shows only the demo project.



Intro to Cloud “composite app”

Example “composite application”



Resources:

Load balancer
  VIP
  Firewall rule for HTTP (port 80)
LB_pool
  Web Server(s) (*small* VMs)

App Servers (*medium* VMs)
    Firewall rule for SSH (port 22)
    Floating IP for each

DB Servers (*large* VMs)

Other Servers (VMs)

Auto-scaling
  What? Web Servers
  How many to add/remove?
  When? Ie;
    CPU > 85% then scale up
    CPU < 15% then scale down