

OpenStack Labs

cloudops.com @cloudops_

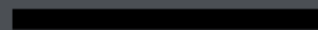
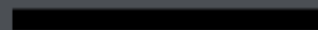
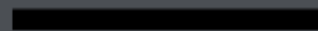


Goals

- Familiarize yourself with Horizon, the OpenStack UI
- Empowering you with the freedom of Self-Provisioning
- Introduce you to the OpenStack CLI



Lab Environment



For Class Use Only - CloudOps



- **Get Your IP** (Put your name beside one please)
 - <https://etherpad.openstack.org/p/cloudops-openstack-workshop>
- **With your browser of choice, please go to:**
 - https://Your_OSA-AiO_IP
 - Accept Self-Signed certificate
 - User/Password: demo / demo





- **Get Your Admin credentials for Horizon**

- Get the id_workshop private key
 - <https://etherpad.openstack.org/p/cloudops-openstack-workshop>
 - Copy everything from ----BEGIN to ----END
 - -----BEGIN RSA PRIVATE KEY-----
blablabla
-----END RSA PRIVATE KEY----
 - Save it into a local file called **id_workshop**

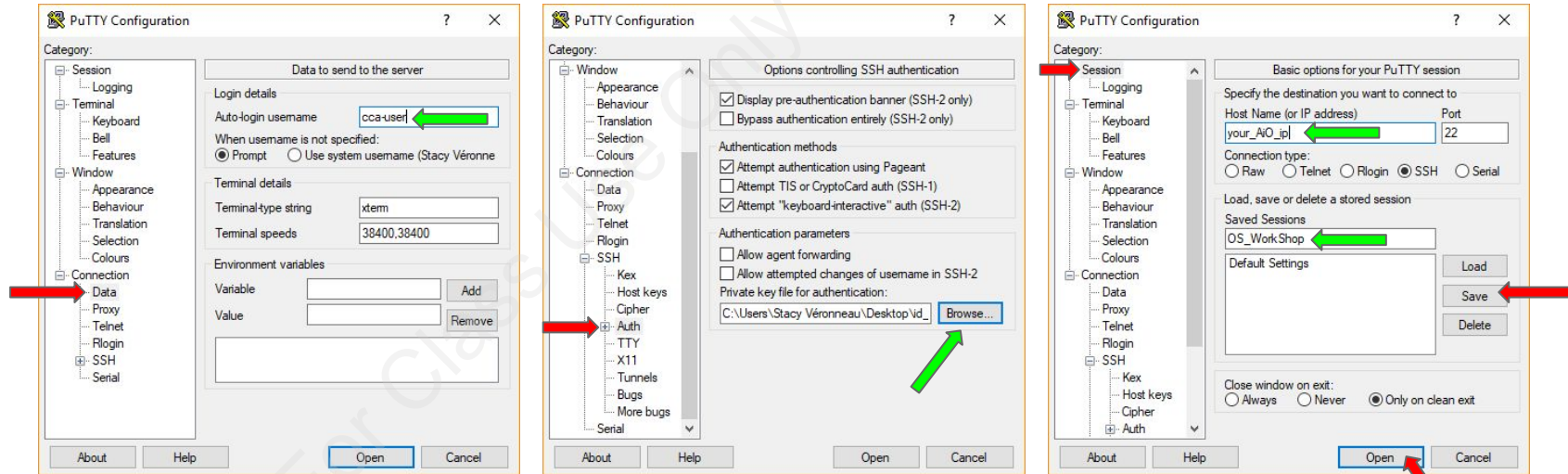




- Get Your *Admin* credentials for Horizon

- PuTTY users:

https://github.com/sveronneau/openstack-ws/blob/master/id_workshop.ppk





- **Get Your *Admin* credentials for Horizon**

- **CLI users:**

- > chmod 0400 id_workshop
- > ssh -i id_workshop cca-user@*Your_AiO_IP*
- > sudo lxc-ls | grep aio1_utility_container-
- > sudo lxc-attach -n aio1_utility_container-**xyzabc123**
- > cat /root/openrc | grep OS_PASSWORD





- **Get Your *Admin* credentials for Horizon**

- From your browser, logout and log back in with this newly acquired power user magic!
 - User/Password: admin / *OS_PASSWORD*

Lab Environment - Is Everybody Ok?



Let's Get Started



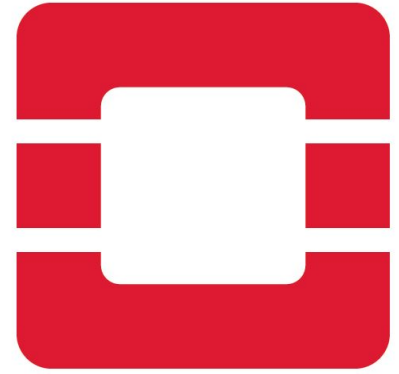
Lab Building 101 - How was this wonderful lab built?



Per Student:

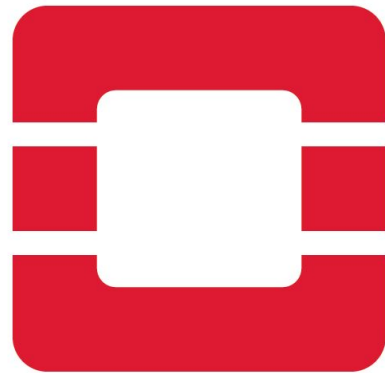
- 1 instance running in **Cloud.ca**
 - Ubuntu 16.04.03
 - 8 vCPUs / 16 GB Ram
 - 25 GB OS + 100 GB Volume
 - 1 nic / 1 Static NAT
 - OpenStack Ansible

Horizon Navigation



For Class Use Only - CloudOps

Spin Instance



For Class Use Only - CloudOps



- **Spin TestVM**

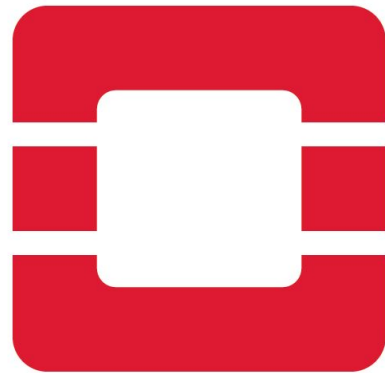
- Project / Compute / Instances / Launch Instance
 - Select name, image, flavor, Networks ... :(ul> - We must make it shareable (Admin / Networks, Edit)
- Start again

- **Console Access**

- Project / Compute / Instances / Actions / Console
 - cirros / cubswin:)



Spin Instance With SSH Keys





- **Spin TestVM**

- Project / Compute / Instances / Launch Instance
 - Select Flavors, Images, Networks, Key Pair
 - Create Key Pair and give it the name **MyKeyPair**
 - Copy Key to clipboard and save it locally under **MyKeyPair.key**





- **Assign Floating IP**

- Project / Compute / Instances / Actions / Associate Floating IP
 - IP Address
 - (+) / public / Allocate IP
 - Port
 - Your instance
 - Associate



Spin Instance With SSH Keys



- **Open your local MyKeyPair.key file**
 - In your editor of choice
- **SSH to your AiO instance**
 - `> sudo lxc-attach -n aio1_utility_container-xyzabc123`
 - `> cd /root`
 - `vim .ssh/MyKeyPair.key`
 - Paste content of local **MyKeyPair.key** and save it
 - `> chmod 0400 .ssh/MyKeyPair.key`



- **Access instance via SSH and Ping**

- > ssh -i .ssh/**MyKeyPair.key** cirros@*instance_fip*
- > ping *instance_fip*
- :(

- **Let's fix that with Security Groups!**

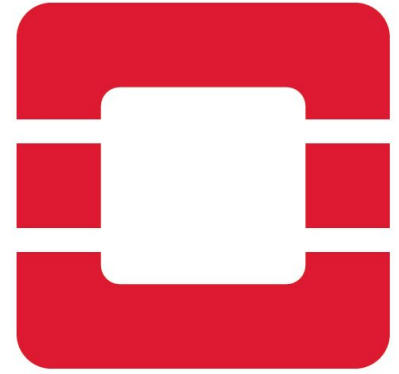
- Project / Network / Security Groups / default (Manage Rules)
 - Add Rule (SSH)
 - Add Rule (All ICMP - Ingress)





- **Access instance via SSH and Ping - Take 2**
 - SSH to your AiO instance
 - ping *instance_fip*
 - ssh -i .ssh/**MyKeyPair.key** cirros@*instance_fip*
 - :)

Projects, Users and Domains





- **Create Project (Admin)**

- Start with the user (Identity / Users)
 - User Name / Password : my_user
 - Primary Project (+)
 - Name : my_project
 - Members and Role (add demo user)
 - Groups (Group of users instead of 1 by 1 members)
 - Quotas
 - Create Project
 - Role = Admin





- **Create Project (Admin)**

- Add my_user to demo project (Identity / Projects)
 - Demo (Manage Members)
 - my_user (member)
- Logout and Log back in with demo user
 - You'll see 2 projects now and you are admin in the new one only.
 - demo - Regular user
 - my_project - Admin



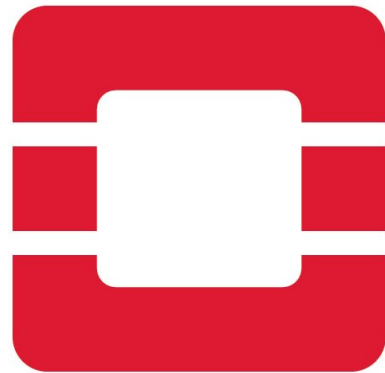


- **Domains (Admin)**

- Introduced with Keystone v3
- Brings an extra level of 'belonging' for a user
- Ties to your LDAP if used for RBAC



Flavors



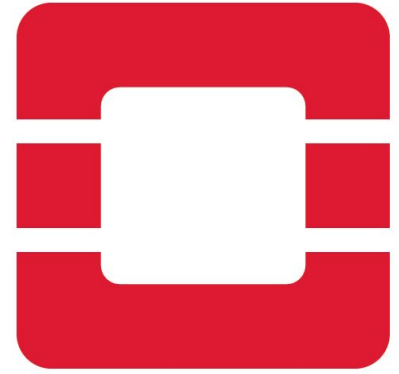
For Class Use Only - CloudOps



- **Let's go back to our admin user**
- **Admin / Compute / Flavors**
 - Create Flavor
 - Info
 - Access
 - Flavors are a set of values and metadata defining minimum requirements.
 - An instance will be shown flavors that matches its minimum requirements



Storage



For Class Use Only - CloudOps



- **Images** (Admin)

- Admin / System / Images

- Download image from

- <https://cloud-images.ubuntu.com/xenial/current/xenial-server-cloudimg-amd64-disk1.img>

- Create Image

- Name / Description
 - Source Type: File
 - Format: QCOW2 (CEPH backend = Use RAW Format)
 - Reason: Copy-On-Write Clone VS QCOW2 copy + extract
 - Metadata





- **Object**

- Project / Object Store / Containers

- +Container

- Name

- Access

- Folder

- Upload File

- <https://royalultimate.com/wp-content/uploads/2016/03/CloudOps-RGB-Stacked-1.png>

- Private / Public (Link)





- **Block**

- Ephemeral VS Persistent
- Project / Volumes / Volumes
 - Create
 - My_volume / 1gb
 - Accept transfer (project-to-project)
 - Edit Volume / Actions
 - Manage Attachments
 - Attach to Instance





- **Block**

- Mount inside instance
 - SSH to instance
 - > `sudo mkdir /MyVolume`
 - > `lsblk`
 - > `sudo mkfs.ext4 /dev/vdb`
 - > `sudo mount -t ext4 /dev/vdb /MyVolume`
 - > `sudo df -kh`



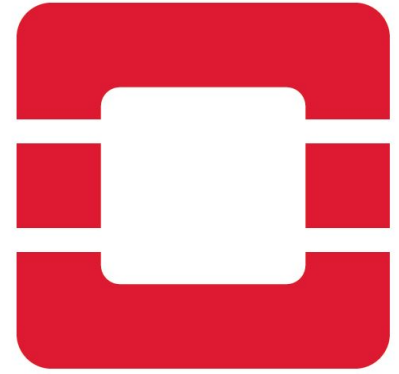


- **Block**

- Project / Compute / Volumes Snapshots
 - Point in time
 - Can be done live
 - Can be used to start a new instance
- Project / Compute / Volumes Backups
 - Full
 - Only from unattached volume
 - Stored in Object Store



Networking



For Class Use Only - CloudOps



- **Neutron 101 recap**

- A logical to physical, API driven network orchestrator (Mini-SDN)

- **Show Networks**

- Default values with OSA

- **Show Routers**

- Default values with OSA

- **Show Subnets**

- Default values with OSA





- **Fix Private Subnet**

- Admin/ System / Networks / private / Subnets / private-subnet

- Edit Subnet

- Subnet Details

- DNS Name Servers

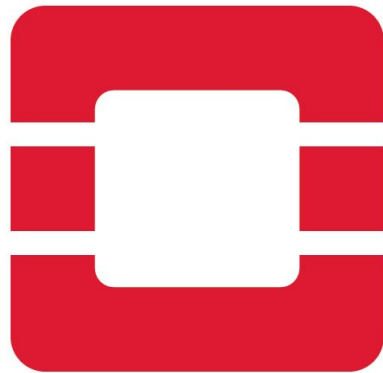
- 8.8.8.8

- 8.8.4.4



Load Balancers

Project Octavia



For Class Use Only CloudOps



- **Make sure you have 2 instances running without floating IPs**
- **Project / Network / Load Balancers**
 - Create Load Balancer
 - LB Details
 - Name + Subnet (private)
 - Listener Details
 - Protocol + Port (tcp / 22)
 - Pool Details
 - Method (round robin)





- **Project / Network / Load Balancers**

- Create Load Balancer

- Pool Members

- Add your instances
 - Port (22)

- Monitor Details

- Type = PING





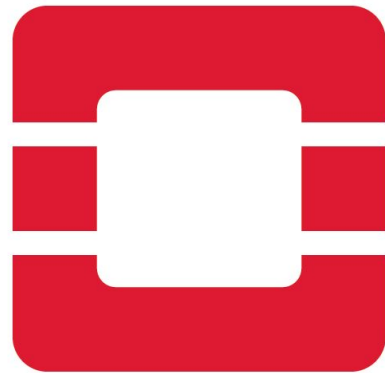
- **Project / Network / Load Balancers**

- Test Load Balancer

- Attach Floating IP to LB (might need to refresh page)
- SSH to AiO
 - `ssh-keygen -f "/root/.ssh/known_hosts" -R <floating ip lb>`
 - `ssh cirros@<floating ip lb>`
 - `> uname -a` Look at host name
 - `> exit`
 - Repeat
 - `ssh-keygen -f "/root/.ssh/known_hosts" -R <floating ip lb>`
 - `ssh cirros@<floating ip lb>`
 - `> uname -a`



Heat



For Class Use Only - CloudOps



- **Heat Templates**

- Provides orchestration and auto-scaling based on HOT templates
 - <https://docs.openstack.org/heat/latest/>



- **Heat Templates**

- Basic (spin a instance, creates network and attach a Floating IP)

- https://github.com/sveronneau/Heat-Templates/blob/master/quick_instance_floating.yaml
- > openstack network list ; openstack image list ; openstack flavor list ; openstack keypair list
 - > vim quick_instance_floating.yaml
- > openstack stack create -t quick_instance_floating.yaml --parameter public_network_id=***\$PUB NETWORK ID*** MyFirstStack
- > openstack stack show MyFirstStack
- Go in Horizon (Project / Orchestration / Stack)



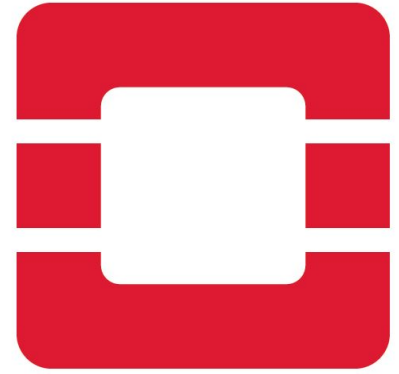
- **Heat Templates**

- Auto-Scaling

- https://github.com/sveronneau/Heat-Templates/blob/master/quick_instance_autoscale.yaml



Other Options

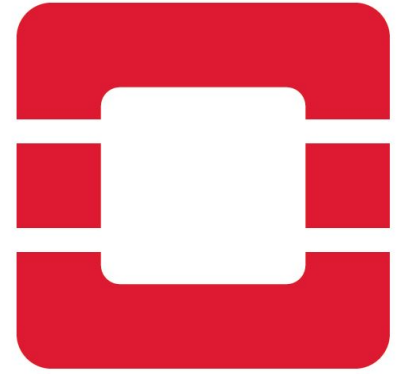


For Class Use Only CloudOps



- **Host Aggregate (Admin) -- Horizontal (Going across racks)**
 - Grouping of physical compute resources
 - Use Metadata to tailor an HA identity (GPU capable ; Faster CPU)
- **Availability Zone (Users) -- Vertical (1 AZ per rack)**
 - Default is Nova
 - Logical representation of HA to users

OpenStack API and CLI





- **API**

- Project / API Access

- **CLI**

- Project / API Access
 - OpenStack clouds.yaml File
 - OpenStack RC File v2.0 (legacy)
 - OpenStack RC File v3.0





- **CLI**

- Download v3.0 file and open it in a text editor
- SSH to your AiO environment LXC Utility Container
 - > cd /root
 - > vim admin-openrc.sh
 - Paste V3.0 local file content and save it
 - > chmod 755 admin-openrc.sh ; cat openrc | grep OS_PASSWORD
 - > ./admin-openrc.sh
 - > openstack image list ; openstack hypervisor list
 - > openstack service list ; openstack endpoint list





- **CLI**

- exit aio1-utility-container
- SSH to your AiO environment LXC Utility Container
 - > cd ; mkdir /home/cca-user/.config ; mkdir /home/cca-user/.config/openstack
 - > cp /root/.config/openstack/clouds.yaml /home/cca-user/.config/openstack





- **CLI**

- > export OS_CLOUD=default
- > openstack image list ; openstack hypervisor list
- > openstack service list ; openstack endpoint list



- **CLI**

- Launch instance from image via CLI

- Step 1 - Gather your parameters (IDs)

- > openstack flavor list ; openstack image list ; openstack security group list ;
openstack keypair list ; openstack network list





- CLI

- Launch instance from image via CLI

- Step 2 - Launch Instance

- > **openstack server create** --flavor *tempest1* --image *cirros* --security-group *458cf86d-6c5d-4ad7-af20-0c4d9e882531* --key-name *MyKeyPair* --nic net-id=*ace8814e-23ec-41e4-8810-6f22cc84d743* *MyLittleServer*

- Step 3 - Look in Horizon





- **Old CLI example**

- `> nova boot --flavor tempest2 --image cirros_alt --security-group 458cf86d-6c5d-4ad7-af20-0c4d9e882531 --key-name MyKeyPair --nic net-id=bd2c7df1-8e9c-4fec-9cd0-0fcb73ab9d05 MyLittleServer2`
 - `:(`
- `> cd /root ; source openrc`
- `> nova boot --flavor tempest2 --image cirros_alt --security-group 458cf86d-6c5d-4ad7-af20-0c4d9e882531 --key-name MyKeyPair --nic net-id=bd2c7df1-8e9c-4fec-9cd0-0fcb73ab9d05 MyLittleServer2`
 - `:)`
- Look in Horizon





- **Install OpenStack Client on your laptop**

- <https://docs.openstack.org/python-openstackclient/latest/>
 - Getting Started section



Thank you for your time



- **Domains (Admin)**

- Identity / Domains / Users / Create Domain
 - Name
 - Enabled
- Introduced with Keystone v3
- Brings an extra level of 'belonging' for a user
- Ties to your LDAP if used for RBAC



Projects, Users And Domains



- **Domains with *LDAP - Jumpcloud.com* Demo (Admin)**
 - Identity / Domains / Create Domain
 - Name / Enabled
- <https://console.jumpcloud.com/login>
- **Back to OpenStack Config Files**
 - Modify Horizon to show domains
 - vi /etc/horizon/local_settings.py
 - OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
 - OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'default'



- **Back to OpenStack Config Files**

- Modify /etc/keystone/keystone.conf to allow per domain configs
- Create /etc/keystone/domains/keystone.acme.com.conf with your LDAP info.

- **Domains with LDAP a *Jumpcloud.com* Demo**

- Login with admin user ; Setup new domain projects; groups; users
- Logout and back in with user from new domain

