nov. 18, 18 17:00	povray-instance.sh	Page 1/1
#!/bin/sh		
openstack container crea openstack server create SecurityGroupkey-name	te povrayContainer flavor m1.tinyimage centos7secu yonidauser-data povray-userdata.sh	rity-group custo povray-instance

```
povray-userdata.sh
 nov. 10, 18 16:44
                                                                             Page 1/1
#!/bin/sh
#Proxy config
export http_proxy=http://proxy.clermont-universite.fr:8080
#openstack intall
apt-get -y install python-openstackclient
#openstack config
export OS_AUTH_URL=http://172.20.88.10:5000/v3
export OS_PROJECT_ID=72fe25d80014433f8bc78a734a5be70b
export OS_PROJECT_NAME="yonidabrah"
export OS_USER_DOMAIN_NAME="Default"
export OS_USERNAME="yonidabrah"
export OS_PASSWORD="Ent12345"
export OS_REGION_NAME="RegionOne"
export OS_INTERFACE=public
export OS_IDENTITY_API_VERSION=3
#Test Config
openstack server list
#Getting zzpovray.tar.gz from container
openstack object save povrayContainer zzpovray.tar.gz
#Extracting povray
tar xfv zzpovray.tar.qz
ls -l
cd povray
#Creating images
./povray +A +W800 +H600 +Lshare/povray-3.6/include/ +SF1 +EF80 glsbng.ini
#uploading images to container
openstack object create povrayContainer *.png
#Shutdown instance
shutdown
```

nov. 18, 18 17:03	post-traitement-instance.sh	Page 1/1
#!/bin/sh		
	-flavor m1.tinyimage centos7security yonidauser-data post-userdata.sh post-t	

```
post-userdata.sh
nov. 10, 18 16:44
                                                                             Page 1/1
#!/bin/sh
#Proxy config
export http_proxy=http://proxy.clermont-universite.fr:8080
#openstack install
apt-get -y install python-openstackclient
#openstack config
export OS_AUTH_URL=http://172.20.88.10:5000/v3
export OS_PROJECT_ID=72fe25d80014433f8bc78a734a5be70b
export OS_PROJECT_NAME="yonidabrah"
export OS_USER_DOMAIN_NAME="Default"
export OS_USERNAME="yonidabrah"
export OS_PASSWORD="Ent12345"
export OS_REGION_NAME="RegionOne"
export OS_INTERFACE=public
export OS_IDENTITY_API_VERSION=3
#Test Config
openstack server list
#ImageMagick install
apt-get -y install imagemagick
#Get images from container
openstack container save povrayContainer
#Create GIF file
convert *.png -delay 6 -quality 100 glsbng.gif
#Upload GIF file to povrayContainer
openstack object create povrayContainer glsbng.gif
#Shutdown instance
shutdown
```

nov. 18, 18 17:05	heat-v1.yaml	Page 1/2
heat_template_version: 201	3-05-23	
<pre>default: yonida image:   type: string</pre>	Reypair to assign to server image to use for server	
type: number	For WaitCondition, depends on your ima	age and environmen
resources:    wait_condition:    type: OS::Heat::WaitCo    properties:     handle: {get_resource     count: 1     timeout: {get_param:	ce: wait_handle}	
wait_handle: type: OS::Heat::WaitCo	onditionHandle	
#openstac apt-get -y ins #openstack cor	<pre>image} flavor} n: key_name} worfig coxy=http://proxy.clermont-universite. ck install stall python-openstackclient unifig</pre>	.fr:8080
export OS_AUTH export OS_PROJ export OS_PROJ export OS_USEF export OS_USEF export OS_PASS export OS_REGI export OS_INTE export OS_IDEN #Test Config openstack serv #Get zzpovray.	H_URL=http://172.20.88.10:5000/v3 JECT_ID=72fe25d80014433f8bc78a734a5be7 JECT_NAME="yonidabrah" R_DOMAIN_NAME="Default" RNAME="yonidabrah" SWORD="Ent12345" ION_NAME="RegionOne" SRFACE=public NTITY_API_VERSION=3 Jer list Ltar.gz from container Lect save povrayContainer zzpovray.tar.	

```
heat-v1.yaml
 nov. 18, 18 17:05
                                                                        Page 2/2
            tar xfv zzpovray.tar.qz
            ls -l
            cd povrav
            #Create images
            ./povray +A +W800 +H600 +Lshare/povray-3.6/include/ +SF1 +EF80 qlsb
ng.ini
            #Upload images to container
            openstack object create povrayContainer *.png
            wc_notify --data-binary '{"status": "SUCCESS"}'
            #Shutdown instance
            shutdown
          params:
            wc_notify: { get_attr: ['wait_handle', 'curl_cli'] }
 genGifInstance:
    type: OS::Nova::Server
    depends_on: wait_condition
    properties:
      image: {get_param: image}
      flavor: {get_param: flavor}
      key_name: {get_param: key_name}
      user data format: RAW
      user_data:
        str_replace:
         template: |
            #!/bin/sh
                 #Proxy config
            export http_proxy=http://proxy.clermont-universite.fr:8080
                 #openstack install
            apt-get -y install python-openstackclient
            #openstack config
            export OS_AUTH_URL=http://172.20.88.10:5000/v3
            export OS_PROJECT_ID=72fe25d80014433f8bc78a734a5be70b
            export OS PROJECT NAME="vonidabrah"
            export OS_USER_DOMAIN_NAME="Default"
            export OS_USERNAME="yonidabrah"
            export OS_PASSWORD="Ent12345"
            export OS REGION NAME="RegionOne"
            export OS_INTERFACE=public
            export OS_IDENTITY_API_VERSION=3
            #Test Config
            openstack server list
            #ImageMagick install
            apt-get -y install imagemagick
            #Get images from container
            openstack container save povrayContainer
            #Create GIF file
            convert *.png -delay 6 -quality 100 glsbng.gif
                 #Upload GIF file to povrayContainer
            openstack object create povrayContainer glsbng.gif
                 #Shutdown instance
            shutdown
          params:
            wc_notify: { get_attr: ['wait_handle', 'curl_cli'] }
outputs:
 curl_cli:
   value: { get_attr: ['wait_handle', 'curl_cli'] }
 wc data:
    value: { get_attr: ['wait_condition', 'data'] }
```

```
heat-v2.vaml
nov. 18, 18 17:05
                                                                        Page 1/3
heat_template_version: 2013-05-23
parameters:
 kev name:
   type: string
   description: Name of keypair to assign to server
   default: vonida
 image:
   type: string
   description: Name of image to use for server
   default: centos7
 flavor:
   type: string
   description: Flavor to use for server
   default: ml.tiny
 timeout:
   type: number
   description: Timeout for WaitCondition, depends on your image and environmen
   default: 800
 numLots:
   type: number
   description: number of images per lot
   default: 60
 indexList:
   type: comma_delimited_list
   label: start index of lots
   default: "1"
 numberLots:
   type: number
   label: number of lots
   default: 1
resources:
 wait condition:
   type: OS::Heat::WaitCondition
   properties:
     handle: {get resource: wait handle}
     # Note, count of 5 vs 6 is due to duplicate signal ID 5 sent below
      count: {get resource: numberLots}
     timeout: {get_param: timeout}
 wait handle:
   type: OS::Heat::WaitConditionHandle
   for_each:
     <%index%>: {get_param: indexList}
   template:
     povray<%index%>:
       type: OS::Nova::Server
       properties:
          image: {get_param: image}
          flavor: {get_param: flavor}
          key_name: {get_param: key_name}
          user_data_format: RAW
         user_data:
            str_replace:
             template:
                #!/bin/sh
                        #Proxy config
                export http_proxy=http://proxy.clermont-universite.fr:8080
```

```
heat-v2.yaml
nov. 18, 18 17:05
                                                                        Page 2/3
                       #openstack install
                apt-get -y install python-openstackclient
                #openstack config
               export OS AUTH URL=http://172.20.88.10:5000/v3
               export OS_PROJECT_ID=72fe25d80014433f8bc78a734a5be70b
               export OS PROJECT NAME="vonidabrah"
               export OS USER DOMAIN NAME="Default"
               export OS USERNAME="vonidabrah"
               export OS_PASSWORD="Ent12345"
               export OS REGION NAME="RegionOne"
               export OS_INTERFACE=public
               export OS IDENTITY API VERSION=3
               #Test Config
               openstack server list
               #Get zzpovray.tar.gz from container
               openstack object save povrayContainer zzpovray.tar.gz
               #Extract povray
               tar xfv zzpovray.tar.qz
               ls -l
               cd povray
               #Creating images
               ./povray +A +W800 +H600 +Lshare/povray-3.6/include/ +SF'ind' +EF
'ind + nmLot' glsbng.ini
               #Upload images to container
               openstack object create povrayContainer *.png
               wc_notify --data-binary '{"status": "SUCCESS"}'
               #Shutdown instance
               shut.down
             params:
               wc_notify: { get_attr: ['wait_handle', 'curl_cli'] }
               ind: <%index%>
               nmLot: {get_param: numLots}
 genGifInstance:
   type: OS::Nova::Server
   depends_on: wait_condition
   properties:
     image: {get param: image}
     flavor: {get_param: flavor}
     key_name: {get_param: key_name}
     user data format: RAW
     user_data:
       str replace:
         template: |
           #!/bin/sh
           export http_proxy=http://proxy.clermont-universite.fr:8080
           apt-get -y install python-openstackclient
           #openstack config
           export OS_AUTH_URL=http://172.20.88.10:5000/v3
           export OS_PROJECT_ID=72fe25d80014433f8bc78a734a5be70b
           export OS_PROJECT_NAME="yonidabrah"
           export OS_USER_DOMAIN_NAME="Default"
           export OS USERNAME="vonidabrah"
           export OS_PASSWORD="Ent12345"
           export OS_REGION_NAME="RegionOne"
           export OS_INTERFACE=public
           export OS_IDENTITY_API_VERSION=3
           #Test Config
           openstack server list
           #install ImageMagick
           apt-get -y install imagemagick
           #Get images from container
```

```
heat-v2.yaml
nov. 18, 18 17:05
                                                                             Page 3/3
            openstack container save povrayContainer
            #Create GIF file
            convert *.png -delay 6 -quality 100 glsbng.gif
#Upload GIF file to povrayContainer
            openstack object create povrayContainer glsbng.gif
                  #Shutdown instance
            shutdown
          params:
            wc_notify: { get_attr: ['wait_handle', 'curl_cli'] }
outputs:
 curl_cli:
   value: { get_attr: ['wait_handle', 'curl_cli'] }
 wc_data:
   value: { get_attr: ['wait_condition', 'data'] }
```