CHAPTER 4

System Implementation

The primary focus of this document is the integration between Cisco PSC and Red Hat OpenShift PaaS. Most of this work involves development of PSC extensions leveraging PSC Request Center and ServiceLink components. Almost no changes occurred on the OpenShift side because the majority of management and monitoring operations are exposed via Restful management API on OpenShift Broker.

For this testbed, a fully built and configured CIAC 3.1.1 virtual appliance running Windows 2008R2 is used. Active Directory is set up and configured as part of the virtual appliance.

For the infrastructure, a fully populated UCS 5108 chassis connected to UCS 6248UP Fabric Interconnects are used. Up links on FIs are connected to the Nexus 5K switch. Two VLANs are used to the servers group—a management VLAN with subnet 172.29.87.224/28 and a public VLAN with subnet 172.29.87.240/28.

Note User familiarity with UCS hardware configuration via UCS Manager is required so documents that detail UCS Manager configurations only are referenced.

Red Hat OpenStack Deployment

The following configurations were used in Red Hat OpenStack testbed deployment.

Controller Node Configuration

The following instructions provide guidance for configuring controller node.

The controller node is assigned to node01.ctocllab.cisco.com (172.29.87.229).

Configure the network interfaces on controller node:

```
Interface eth0: ifcfg-eth0

DEVICE=eth0
TYPE=Ethernets
ONBOOT=yes
NM_CONTROLLED=no
```

```
BOOTPROTO=static
IPADDR=172.29.87.229
PREFIX=28
GATEWAY=172.29.87.225
DNS1=172.29.74.154
DOMAIN=ctocllab.cisco.com
DEFROUTE=yes

 Interface eth1: ifcfg-eth1
DEVICE=eth1
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=none
PROMISC=yes
```

Installing and Configuring Controller Node

The following instructions provide guidance for installing and configuring controller node.

The controller node is assigned to [node01.ctocllab.cisco.com (172.29.87.229)].

Register system with Red Hat Network for a pre-registered user id.

```
#> subscription-manager register
```

List available or consumed subscriptions for registered system.

```
#> subscription-manager list [--available|--consumed]
```

Attach the system to a given Pool Id for RHOS 4 (Havana).

```
Subscription Name: Red Hat Cloud Infrastructure Business Partner Self-Supported NFR
(4-sockets)
Provides:          Red Hat OpenStack
                   JBoss Enterprise Application Platform
                   Red Hat Enterprise Linux Server
                   Red Hat OpenStack Beta
                   Red Hat Enterprise Virtualization
                   Red Hat Enterprise MRG Messaging 2
                   Red Hat CloudForms
                   Red Hat Enterprise Linux 7 Public Beta
                   Red Hat Beta
SKU:               ...
Contract:          ...
Account:           ...
Serial:            ...
Pool ID:           <pool-id>

#> subscription-manager attach --pool=<pool-id>
```

Install yum-utils to enable relevant openstack rpms for openstack-4.0.

```
#> yum install -y yum-utils
#> yum install -y yum-plugin-priorities
#> yum-config-manager --enable rhel-6-server-openstack-4.0-rpms \
       --setopt="rhel-6-server-openstack-4.0-rpms.priority=1"
#> yum update -y
#> reboot
```

Install and configure the Database server.

```
#> yum install -y mysql-server
#> service mysqld start
```

```
#> chkconfig mysqld on
```

Configure firewall to allow tcp traffic for msql (add the following line to /etc/sysconfig/iptables).

```
"-A INPUT -p tcp -m multiport --dports 3306 -j ACCEPT"
#> service iptables restart
```

Set the database administrator password.

```
#> /usr/bin/mysqladmin -u root password "PASSWORD"
```

Install and configure the Message Broker (qpid).

```
#> yum install -y qpid-cpp-server qpid-cpp-server-ssl
```

Register qpid users.

```
#> saslpasswd2 -f /var/lib/qpidd/qpidd.sasldb -u QPID cinder
#> saslpasswd2 -f /var/lib/qpidd/qpidd.sasldb -u QPID neutron
#> saslpasswd2 -f /var/lib/qpidd/qpidd.sasldb -u QPID nova
#> sasldblistusers2 -f /var/lib/qpidd/qpidd.sasldb
```

Configure firewall to allow tcp traffic for qpid (add the following line to /etc/sysconfig/iptables).

```
"-A INPUT -p tcp -m tcp --dport 5672  -j ACCEPT"
#> service iptables restart
```

Start the service.

```
#> service qpidd start
#> chkconfig qpidd on
```

Installing OpenStack Identity

The following instructions provide guidance for installing the OpenStack identity service.

Keystone Configuration (/etc/keystone/keystone.conf).

```
[DEFAULT]
admin_token = 12de4ec0f1a924f3b20e
bind_host = 172.29.87.229
compute_port = 8774
debug = False
verbose = False
[sql]
connection = mysql://keystone:<passwd>@172.29.87.229/keystone
[identity]
driver = keystone.identity.backends.sql.Identity
[catalog]
driver = keystone.catalog.backends.sql.Catalog
[token]
driver = keystone.token.backends.sql.Token
[signing]
token_format = UUID
[auth]
methods = password,token
password = keystone.auth.plugins.password.Password
token = keystone.auth.plugins.token.Token
```

Certain python package dependencies for openstack-keystone need to be explicitly installed.

```
#> yum install python-setuptools
#> easy_install pip
#> pip install six
#> pip install sqlalchemy
```

Configure firewall (add the following line to /etc/sysconfig/iptables).

```
"-A INPUT -p tcp -m multiport --dports 5000,35357 -j ACCEPT"
#> service iptables restart
```

Installing OpenStack Image Service

The following instructions provide guidance for installing OpenStack image service.

Glance configuration

(/etc/glance/glance-api.conf)

```
[DEFAULT]
default_store = file
bind_host = 0.0.0.0
bind_port = 9292
backlog = 4096
sql_idle_timeout = 3600
workers = 1
registry_host = 0.0.0.0
registry_port = 9191
registry_client_protocol = http

qpid_notification_exchange = glance
qpid_notification_topic = notifications
qpid_hostname = localhost
qpid_port = 5672
qpid_username =
qpid_password =
qpid_sasl_mechanisms =
qpid_reconnect_timeout = 0
qpid_reconnect_limit = 0
qpid_reconnect_interval_min = 0
qpid_reconnect_interval_max = 0
qpid_reconnect_interval = 0
qpid_protocol = tcp
qpid_tcp_nodelay = True

delayed_delete = False

scrub_time = 43200
sql_connection = mysql://glance:<passwd>@172.29.87.229/glance

[keystone_authtoken]
auth_host = 172.29.87.229
auth_port = 35357
auth_protocol = http
admin_tenant_name = services
admin_user = glance
admin_password = <passwd>
[paste_deploy]
flavor = keystone

(/etc/glance/glance-registry.conf)
[DEFAULT]
bind_host = 0.0.0.0
bind_port = 9191
backlog = 4096
sql_idle_timeout = 3600
api_limit_max = 1000
limit_param_default = 25
```

```
sql_connection = mysql://glance:<passwd>@172.29.87.229/glance
[keystone_authtoken]
auth_host = 172.29.87.229
auth_port = 35357
auth_protocol = http
admin_tenant_name = services
admin_user = glance
admin_password = <password>
[paste_deploy]
flavor = keystone
```

Configure firewall (add the following line to /etc/sysconfig/iptables).

```
"-A INPUT -p tcp -m multiport --dports 9292 -j ACCEPT"
#> chown -R glance:glance /var/log/glance/registry.log
```

Verify Glance service installation.

Download image and add to glance.

```
#> wget http://cdn.download.cirros-cloud.net/0.3.1/cirros-0.3.1-x86_64-disk.img
#> glance image-create --name="CirrOS 0.3.1" --disk-format=qcow2
--container-format=bare --is-public=true < cirros-0.3.1-x86_64-disk.img
#> glance image-list
+--------------------------------------+---------------------------------+------------
-+------------------+-------------+--------+
| ID                                   | Name                            | Disk Format
| Container Format | Size        | Status |
+--------------------------------------+---------------------------------+------------
-+------------------+-------------+--------+
| e71d8b33-d737-47d3-b5ed-32b085d0b47f | CirrOS 0.3.1                    | qcow2
| bare             | 13147648    | active |
+--------------------------------------+---------------------------------+------------
-+------------------+-------------+--------+
```

Installing OpenStack Compute Service

Refer to Installing the OpenStack Compute Service.

Install and Configure VNC Proxy.

```
#> yum install -y openstack-nova-novncproxy
#> yum install -y openstack-nova-console

(/etc/nova/nova.conf)
novncproxy_host = 172.29.87.229
novncproxy_port=6080
novncproxy_base_url=http://172.29.87.229:6080/vnc_auto.html
vncserver_listen=0.0.0.0
vnc_enabled=true
vnc_keymap=en-us
```

Configure firewall (add the following line to /etc/sysconfig/iptables).

```
"-A INPUT -m state --state NEW -m tcp -p tcp --dport 6080 -j ACCEPT"
#> service iptables restart
#> service openstack-nova-consoleauth start
#> chkconfig openstack-nova-consoleauth on
#> service openstack-nova-novncproxy start
#> chkconfig openstack-nova-novncproxy on
```

Create Compute Service Database and Identity Records.

Install Compute packages.

```
#> yum install -y openstack-nova-api \
   openstack-nova-conductor openstack-nova-scheduler \
   python-cinderclient
* Change file ownership:
#> chown -R root:nova api-paste.ini
#> chown -R root:nova nova.conf
#> chown -R root:nova policy.json
#> chown -R root:nova rootwrap.conf
#> chown -R nova:nova /var/log/nova/nova-api.log
* Start messagebus
#> service messagebus start
#> service messagebus status
#> chkconfig messagebus on

(/etc/nova/nova.conf)
[DEFAULT]
rpc_backend = nova.openstack.common.rpc.impl_qpid
my_ip = 172.29.87.229
auth_strategy =keystone
sql_connection = mysql://nova:<passwd>@172.29.87.229/nova
enabled_apis=ec2,osapi_compute
osapi_compute_listen=172.29.87.229
osapi_compute_listen_port=8774
metadata_listen=0.0.0.0
metadata_listen_port=8700
api_paste_config=/etc/nova/api-paste.ini
service_neutron_metadata_proxy=True
neutron_metadata_proxy_shared_secret=cisco123
novncproxy_host = 172.29.87.229
novncproxy_port=6080
network_api_class=nova.network.neutronv2.api.API
metadata_host=172.29.87.229
neutron_url=http://172.29.87.229:9696
neutron_admin_username=neutron
neutron_admin_password=cisco123
neutron_admin_tenant_name=services
neutron_admin_auth_url=http://172.29.87.229:35357/v2.0
security_group_api=nova
debug=true
qpid_hostname = 172.29.87.229
qpid_username = nova
qpid_password = cisco123
qpid_port = 5672
scheduler_default_filters=AllHostsFilter
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
allow_same_net_traffic=true
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
novncproxy_base_url=http://172.29.87.229:6080/vnc_auto.html
vncserver_listen=0.0.0.0
vnc_enabled=true
vnc_keymap=en-us
[keystone_authtoken]
auth_host = 172.29.87.229
auth_protocol = http
auth_port = 35357
admin_user = nova
admin_tenant_name = services
admin_password = <password>
```

Compute Services to start and chkconfig.

```
#> service openstack-nova-api restart
#> service openstack-nova-conductor restart
#> service openstack-nova-console restart
```

```
#> service openstack-nova-consoleauth restart
#> service openstack-nova-metadata-api restart
#> service openstack-nova-novncproxy restart
#> service openstack-nova-scheduler  restart
#> chkconfig openstack-nova-api on
#> chkconfig openstack-nova-conductor on
#> chkconfig openstack-nova-console on
#> chkconfig openstack-nova-consoleauth on
#> chkconfig openstack-nova-metadata-api on
#> chkconfig openstack-nova-novncproxy on
#> chkconfig openstack-nova-scheduler  on
```

Openstack Block Storage.

Install and Configure Block Storage.

Volume Service Specific Configuration: The block storage driver used in this configuration is LVM. It uses a file mounted via a loop device where a LVM has been created.

Create a new logical volume.

```
#> lvcreate -L 800G -n lv_vol_ephemeral vg_node01
  lv_vol_ephemeral                              vg_node01      -wi-ao---- 800.00g
#> mkfs -t ext4 /dev/vg_node01/lv_vol_ephemeral
#> mkdir /os_scratch
```

Add entry in /etc/fstab.

```
"/dev/vg_node01/lv_vol_ephemeral /os_scratch     ext4    defaults        1 3"
#> mount /os_scratch
#> dd if=/dev/zero of=/os_scratch/cinder-volumes bs=1 count=0 seek=800G
#> losetup -fv /os_scratch/cinder-volumes
```

Check the loop device associated with /os_scratch/cinder-volumes.

```
#> losetup -a
/dev/loop0: [fd02]:13 (/os_scratch/cinder-volumes)
```

Create the volume group associated with it.

```
#> vgcreate cinder-volumes /dev/loop0
#> vgs
  VG             #PV #LV #SN Attr   VSize   VFree
  cinder-volumes   1   4   0 wz--n- 800.00g 715.00g
  vg_node01        1   3   0 wz--n- 931.02g  77.02g

#> echo "include /etc/cinder/volumes/*" >> /etc/tgt/targets.conf
#> yum install scsi-target-utils
#> service tgtd start
#> chkconfig tgtd on
```

(/etc/cinder/cinder.conf).

```
[DEFAULT]
auth_strategy = keystone
rpc_backend = cinder.openstack.common.rpc.impl_qpid
qpid_hostname = 172.29.87.229
qpid_username = cinder
qpid_password = <passwd>
sql_connection = mysql://cinder:<passwd>@172.29.87.229/cinder
service_down_time=180
volume_group=cinder-volumes
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
[keystone_authtoken]
auth_host = 172.29.87.229
admin_tenant_name = services
```

```
admin_user = cinder
admin_password = <passwd>
```

Validate the setup.

```
#> cinder create --display-name vol-test 1
+---------------------+--------------------------------------+
|       Property      |                Value                 |
+---------------------+--------------------------------------+
|     attachments     |                  []                  |
|  availability_zone  |                 nova                 |
|       bootable      |                false                 |
|      created_at     |      2014-02-06T17:42:16.996571      |
| display_description |                 None                 |
|     display_name    |               vol-test               |
|          id         | dea6e442-69af-4925-bb58-4decf301a13e |
|       metadata      |                  {}                  |
|         size        |                  1                   |
|     snapshot_id     |                 None                 |
|     source_volid    |                 None                 |
|        status       |               creating               |
|     volume_type     |                 None                 |
+---------------------+--------------------------------------+
#> lvs
  LV                                          VG             Attr       LSize   Pool
Origin Data%  Move Log Cpy%Sync Convert
  volume-dea6e442-69af-4925-bb58-4decf301a13e cinder-volumes -wi-ao----   1.00g
  ....
```

Installing and Configuring Neutron Services

The neutron plugin used in this configuration is Open vSwitch (openstack-neutron-openvswitch). L3 agent abstracts the router that can connect to provide gateway services for L2 networks. The Compute node in this configuration hosts the network services for L3 agent, DHCP agent, and the neutron metadata agent that proxies to the nova metadata service. The neutron server is hosted on the Controller node and the L3-agent, L2-agent, DHCP agent and Metadata agent run on the compute node.

Installing Networking Pre-requisites on the Controller

The following instructions provide guidance for installing networking pre-requisites on the controller.

[Create the Openstack Networking Databasel[https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/Installation_and_Configuration_Guide/sect-Networking_Prerequisite_Configuration.html]]

[Create the Networking identity Records|https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/4/html/Installation_and_Configuration_Guide/Creating_the_Service_Endpoint.html]

Common Networking Configuration.

Disable Network Manager.

Install the relevant packages.

```
#> yum install -y openstack-neutron \
   openstack-utils \
   openstack-selinux
```

Configure the firewall (add entry to /etc/sysconfig/iptables).

```
"-A INPUT -p tcp -m multiport --dports 9696 -j ACCEPT"
```

```
#> service iptables restart
```

(/etc/neutron/neutron.conf).

```
[DEFAULT]
auth_strategy = keystone
rpc_backend = neutron.openstack.common.rpc.impl_qpid
qpid_hostname = 172.29.87.229
qpid_username = neutron
qpid_password = <passwd>
core_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
ovs_use_veth = True
allow_overlapping_ips = True
debug = True
dhcp_lease_duration = 604800
dhcp_lease_time = 604800
[quotas]
[agent]
root_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf
[keystone_authtoken]
auth_host = 172.29.87.229
admin_tenant_name = services
admin_user = neutron
admin_password = <passwd>
[database]
[service_providers]
Launch Networking Service
service neutron-server start
chkconfig neutron-server on
```

Installing Horizon Dashboard

The following instructions provide guidance for installing Horizon Dashboard.

```
#> yum install -y mod_wsgi httpd
#> yum install -y memcached python-memcached
#> yum install -y openstack-dashboard
#> service httpd start
#> chkconfig httpd on
```

Check if httpd is running.

```
#> service --status-all | grep httpd
```

Configure connections and logging.

Edit /etc/openstack-dashboard/local_settings.

Configure local memory cache settings.

```
CACHES = {
    'default': {
        'BACKEND' : 'django.core.cache.backends.locmem.LocMemCache'
    }
}
OPENSTACK_HOST = "172.29.87.229"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v2.0" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "admin"
```

Restart httpd service.

```
#> service httpd restart
```

Configure SELinux.

Allow connections from httpd service to the Identity server if SELinux is configured in 'Enforcing or 'Permissive' mode

```
#> getenforce

Enforcing
#> setsebool -P httpd_can_network_connect on
```

Configure firewall (add the following line to /etc/sysconfig/iptables).

```
"-A INPUT -p tcp --dport 443 -j ACCEPT"
#> service iptables restart
```

Validate dashboard installation: http://172.29.87.229/dashboard.

Install Openstack Orchestration Service.

Install Openstack Telemetry Service (Ceilometer).

Install and Configure Compute and Network Node (node02.ctocllab.cisco.com (172.29.87.230).

Installing and Configuring Compute Node

The following instructions provide guidance for installing and configuring compute node.

Register system with Red Hat Network for a pre-registered userid.

```
#> subscription-manager register
```

List available or consumed subscriptions for registered system.

```
#> subscription-manager list [--available|--consumed]
```

Attach the system to a given Pool Id for RHOS 4 (Havana).

```
Subscription Name: Red Hat Cloud Infrastructure Business Partner Self-Supported NFR
(4-sockets)
Provides:          Red Hat OpenStack
                   JBoss Enterprise Application Platform
                   Red Hat Enterprise Linux Server
                   Red Hat OpenStack Beta
                   Red Hat Enterprise Virtualization
                   Red Hat Enterprise MRG Messaging 2
                   Red Hat CloudForms
                   Red Hat Enterprise Linux 7 Public Beta
                   Red Hat Beta
SKU:               ...
Contract:          ...
Account:           ...
Serial:            ...
Pool ID:           <pool-id>
#> subscription-manager attach --pool=<pool-id>
```

Install yum-utils to enable relevant openstack rpms for openstack-4.0.

```
#> yum install -y yum-utils
#> yum install -y yum-plugin-priorities
#> yum-config-manager --enable rhel-6-server-openstack-4.0-rpms \
       --setopt="rhel-6-server-openstack-4.0-rpms.priority=1"
#> yum update -y
#> reboot
```

Check hardware virtualization support by checking presence of svm or vmx CPU extensions.

```
#> grep -E 'svm|vmx' /proc/cpuinfo
```

Verify kvm modules are loaded.

```
#> lsmod | grep kvm
```

Output must include kvm_intel or kvm_amd.

Check prerequisites.

```
#> yum install ntp
#> service ntpd start
#> chkconfig ntpd on
#> service libvirtd status
libvirtd (pid  2276) is running...
#> service messagebus status
```

Install nova-compute.

```
#> yum install openstack-nova-compute
#> chown root:nova /etc/nova/nova.conf
```

(/etc/nova/nova.conf).

```
[DEFAULT]
rpc_backend = nova.openstack.common.rpc.impl_qpid
my_ip = 172.29.87.230
auth_strategy =keystone
sql_connection = mysql://nova:<passwd>@172.29.87.229/nova
enabled_apis=ec2,osapi_compute,metadata
metadata_listen=0.0.0.0
metadata_listen_port=8775
api_paste_config=/etc/nova/api-paste.ini
service_neutron_metadata_proxy=True
neutron_metadata_proxy_shared_secret=<passwd>
novncproxy_port=6080
glance_host=172.29.87.229
glance_api_servers=$glance_host:$glance_port
network_api_class=nova.network.neutronv2.api.API
metadata_host=$my_ip
metadata_port=8775
neutron_url=http://172.29.87.229:9696
neutron_admin_username=neutron
neutron_admin_password=<passwd>
neutron_admin_tenant_name=services
neutron_admin_auth_url=http://172.29.87.229:35357/v2.0
security_group_api=neutron
debug=true
qpid_hostname = 172.29.87.229
qpid_username = nova
qpid_password = <passwd>
qpid_port = 5672
firewall_driver=nova.virt.firewall.NoopFirewallDriver
allow_same_net_traffic=true
libvirt_vif_driver=nova.virt.libvirt.vif.LibvirtHybridOVSBridgeDriver
novncproxy_base_url=http://172.29.87.229:6080/vnc_auto.html
vncserver_listen=0.0.0.0
vncserver_proxyclient_address=172.29.87.230
vnc_enabled=true
vnc_keymap=en-us
instance_usage_audit = True
instance_usage_audit_period = hour
notify_on_state_change = vm_and_task_state
notification_driver=nova.openstack.common.notifier.rpc_notifier
notification_driver = ceilometer.compute.nova_notifier
[keystone_authtoken]
```

```
auth_host = 172.29.87.229
auth_protocol = http
auth_port = 35357
admin_user = nova
admin_tenant_name = services
admin_password = <password>
```

Start the compute service.

```
#> service openstack-nova-compute start
#> chkconfig openstack-nova-compute on
```

Install and Configure Network.

```
#> yum install openstack-neutron openstack-neutron-openvswitch
#> yum install bridge-utils -y
```

Verify that openvswitch package is installed.

```
#> rpm -qa | grep openvswitch
Start openvswitch service
#> service openvswitch start
#> chkconfig openvswitch on
```

The host running Open vSwitch agent requires that the ovs bridge named br-int be created.

```
#> ovs-vsctl add-br br-int
```

Configure external network access by creating an external bridge.

```
#> ovs-vsctl add-br br-ex
```

The external bridge to the interface on the compute node (ensure it is running in promiscuous mode).

```
Interface eth1 config: ifcfg-eth1
DEVICE=eth1
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=none
PROMISC=yes

#> ovs-vsctl add-port br-ex eth1
```

(/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini)

```
[securitygroup]
firewall_driver =
neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
[OVS]
tenant_network_type = vlan
network_vlan_ranges = physnet1:885:886:887
bridge_mappings = physnet1:br-ex
[DATABASE]
sql_connection = mysql://neutron:<passwd>@172.29.87.229/ovs_neutron
[SECURITYGROUP]
firewall_driver =
neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
/etc/neutron/l3-agent.ini
[DEFAULT]
debug = False
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
ovs_use_veth = True
use_namespaces = True
metadata_ip = 172.29.87.229
metadata_port = 8700
```

```
(/etc/neutron/dhcp-agent.ini)
[DEFAULT]
debug = True
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = True
root_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf

(/etc/neutron/metadata.ini)
[DEFAULT]
debug = True
auth_url = http://172.29.87.229:35357/v2.0
auth_region = regionOne
admin_tenant_name = services
admin_user = neutron
admin_password = <passwd>
nova_metadata_ip = 172.29.87.229
nova_metadata_port = 8700
metadata_proxy_shared_secret = <passwd>

(/etc/neutron/neutron.conf)
[DEFAULT]
auth_strategy = keystone
rpc_backend = neutron.openstack.common.rpc.impl_qpid
qpid_hostname = 172.29.87.229
qpid_username = neutron
qpid_password = <passwd>
core_plugin = neutron.plugins.openvswitch.ovs_neutron_plugin.OVSNeutronPluginV2
ovs_use_veth = True
allow_overlapping_ips = True
debug = True
[quotas]
[agent]
root_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf
[keystone_authtoken]
auth_host = 172.29.87.229
admin_tenant_name = services
admin_user = neutron
admin_password = <passwd>
[database]
[service_providers]
```

Start the network services and chkconfig on.

```
#> service neutron-dhcp-agent start
#> service neutron-l3-agent start
#> service neutron-metadata-agent start
#> service neutron-openvswitch-agent start
#> chkconfig neutron-dhcp-agent on
#> chkconfig neutron-l3-agent on
#> chkconfig neutron-metadata-agent on
#> chkconfig neutron-openvswitch-agent on
```

Red Hat OpenShift Deployment

The ability to spin up OpenShift Node infrastructure on-demand forms the basis for advanced features such as auto-scaling the PaaS infrastructure.

When deployed on OpenStack, OpenShift is able to provision and de-provision Node VMs without manual provisioning steps using Heat templates.

This provides the ability to auto-scale up and down. Currently scaling decisions are made within the broker using platform agnostic scaling scripts. OpenShift applies a push approach in which it initiates the scale events based on the information is has already gathered.

As the Node comes on line and starts the MCollective service, it becomes available to pick up messages from the broker.

Installing the Broker and Node Infrastructure

The following instructions provide guidance for installing the broker and node infrastructure.

Download the latest qcow2 RHEL 6.5 Guest Image from RHN.

Red Hat Common (for RHEL 6 Server x86_64).

Download the latest heat templates for OpenShift Enterprise (Broker and Node).

Use diskimage-builder to prepare the RHEL 6.5 image.

```
#> git clone https://github.com/openstack/diskimage-builder.git
```

The heat templates used to spin up the broker and the node will require enterprise licenses and pool ids for subscriptions to register the systems for the broker and the node at the time of instantiation.

Prepare to run diskimage-builder.

```
#> mkdir $HOME/tmp
```

Export path to the do DIB elements for OpenShift Enterprise.

```
#> export
ELEMENTS_PATH=heat-templates/elements:heat-templates/openshift-enterprise/dib/elements
```

Host the downloaded file on a local httpd server since builder uses an http endpoint to download the image and remote image locations on RHN tend to update the ISO images on frequently.

```
#> export DIB_CLOUD_IMAGES=http://localhost/
#> export DIB_RHSM_OSE_POOL=<ose-pool-id>
#> export DIB_RHSM_POOL=<ose-pool-id>
#> export DIB_RHSM_USER=<registered user>
#> export DIB_RHSM_PASSWORD=<password>
#> export TMP_DIR=$HOME/tmp
#> export DIB_IMAGE_SIZE=10
#> export DIB_OSE_VERSION=2.0
#> export DIB_YUM_VALIDATOR_VERSION=2.0
```

Do not set DIB_RHSM_USER and DIB_RHSM_PASSWORD. It fails during subscription manager register phase. The OSE Pool Id should suffice.

Unit subscriptions do not get removed from systems that fail builds in the midst. The outcome is that at some point, the build will exhaust them and error out with no more subscriptions available from pool.

Remove subscriptions attached to the build server from https://access.redhat.com/management/consumers (All Units)

Bug tracked at https://bugzilla.redhat.com/show_bug.cgi?id=1004483

Run the diskimage-builder for the broker to generate a rhel image using DIBs for openshift-enterprise-broker

```
#> diskimage-builder/bin/disk-image-create --no-tmpfs -a amd64 vm rhel
openshift-enterprise-broker -o RHEL65upgraded-x86_64-broker-v3
```

Run the diskimage-builder for the node to generate a rhel image using DIBs for openshift-enterprise-node.

```
#> diskimage-builder/bin/disk-image-create --no-tmpfs -a amd64 vm rhel
openshift-enterprise-node -o RHEL65-x86_64-node-v2

Upload the Broker image (RHEL65upgraded-x86_64-broker-v3.qcow2) to the target
OpenStack deployment
#> glance image-create --name=RHEL65upgraded-x86_64-broker-v3.qcow2
--disk-format=qcow2 --container-format=bare --is-public=true <
RHEL65upgraded-x86_64-broker-v3.qcow2

Upload the Node image (RHEL65-x86_64-node-v2.qcow2) to the target OpenStack deployment
#> glance image-create --name=RHEL65-x86_64-node-v2 --disk-format=qcow2
--container-format=bare --is-public=true < RHEL65-x86_64-node-v2.qcow2

Console root login has been disabled for the images. Used virt-sysprep to inject the
password at first-boot. The virt-sysprep utility is part of the ibguestfs-tools-c
package.
#> virt-sysprep --firstboot <script-file.sh>  -a <image-filename>

At the end of the image build, the umount2: Invalid argument, umount:
/root/tmp/image.BKl7ZzO2: not mounted error can be ignored
```

Openstack VM boot time console logs do not show up, and needs to be added to grub command line in the image, however grub is currently not enabled on rhel 6.5, but is in progress.

Openstack Metadata service is exclusively used by cloud-init scripts in the broker/node images to run user data passed on during boot time. Enabling config drive has no effect. In fact, the broker wait conditions also exclusively use the Metadata service of Openstack to communicate cloud-init completion.

Automated Deployments using Heat Templates

Spin up the broker and the node instances using subscription manager on a given network with heat cli

```
#> heat create openshift-1
--template-url=http://172.29.87.229/heat-templates/OpenShift-1B1N-neutron-cisco.yaml \
--parameters="key_name=<keypair>;prefix=<domain name>;upstreamDNS=<dns ip address>;\
broker_image_name=RHEL65upgraded-x86_64-broker-v2;node_image_name=RHEL65-x86_64-node-v
2;\
BrokerHostname=ose-broker-1.ctocllab.cisco.com;NodeHostname=ose-node-1.ctocllab.cisco.
com;\
ConfInstallMethod=rhsm;ConfSMRegName=<rhn-register-user>;ConfSMRegPass=<password>;\
ConfSMRegPool=<ose-poold-id>;private_net_id=<uuid-of-internal-neutron-network>;\
public_net_id=uuid-of-external-neutron-networ;private_subnet_id=<uuid-of-internal-neut
ron-subnet>;\
ose_version=2.0;yum_validator_version=2.0"
```

The broker and node will take several minutes to complete cloud-init configuration.

Once completed, access the VM via the VNC console and run oo-diagnostics on the broker and the node.

OpenShift AD Configuration

Perform the following procedure to configure OpenShift AD.

Step 1 Log on to the OpenShift broker server.

Step 2 Edit broker HTTPD configuration file at:

/var/www/openshift/broker/httpd/conf.d/openshift-origin-auth-remote-user.conf

Step 3 Add the following configuration:

```
LoadModule ldap_module modules/mod_ldap.so
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
<Location /broker>
    AuthName "OpenShift"
    AuthType Basic
    AuthBasicProvider ldap
    AuthLDAPURL
"ldap://<AD-SERVER>:389/DC=CISCODEMO,DC=local?sAMAccountName?sub?(objectClass=*)" NONE
    AuthLDAPBindDN "Administrator@ciscodemo.local"
    AuthLDAPBindPassword "PASSWORD"
    require valid-user
```

Step 4 Restart OpenShift broker service:

```
#> service openshift-broker restart
```

Step 5 Edit console configuration file at:
/var/www/openshift/console/httpd/conf.d/openshift-origin-auth-remote-user.conf

Step 6 Add the following configuration:

```
LoadModule ldap_module modules/mod_ldap.so
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
<Location /console>
    AuthName "OpenShift"
    AuthType Basic
    AuthBasicProvider ldap
    AuthLDAPURL
"ldap://<AD-SERVER>:389/DC=CISCODEMO,DC=local?sAMAccountName?sub?(objectClass=*)" NONE
    AuthLDAPBindDN "Administrator@ciscodemo.local"
    AuthLDAPBindPassword "PASSWORD"
    require valid-user
```

Step 7 Restart OpenShift console service:

```
#> service openshift-console restart
```

Creating OpenShift District

Districts define a set of node hosts within which gears can be reliably moved to manage the resource usage of those nodes. While not strictly required for a basic OpenShift installation, their use is recommended where administrators might ever need to move gears between nodes. It's also possible to create multiple districts and designate different security level on each district by leveraging network firewall and access control policies associated with node network address.

To use districts, the broker's MCollective plugin must be configured to enable districts.

Step 1 Edit the /etc/openshift/plugins.d/openshift-origin-msg-broker-mcollective.conf configuration file and confirm the following parameters are set:

```
DISTRICTS_ENABLED=true
NODE_PROFILE_ENABLED=true
DISTRICTS_REQUIRE_FOR_APP_CREATE=true
```

Step 2 Execute the following command on the broker host:

```
#> oo-admin-ctl-district -c create -n smal_district -p small
```

Where small_district is the name of the new district and small is the profile of the gears that will be provisioned on this district.

Step 3 Add node host to small_district that was created:

```
#> oo-admin-ctl-district -c add-node -n small_district -i node.example.com
```

It is important to note that the server identity (node.example.com here) is the node's hostname as configured on that node, which could be different from the PUBLIC_HOSTNAME configured in /etc/openshift/node.conf on the node. The PUBLIC_HOSTNAME is used in CNAME records and must resolve to the host via DNS; the hostname could be something completely different and may not resolve in DNS at all.

CIAC Configuration

CIAC needs to be configured to create services and portals outlined in the design section. The following instructions lay out the necessary configurations.

Active Directory Integration

Perform the following procedure to create integrate active directory.

Step 1 Launch Cisco Service Portal in browser with administration user.

Step 2 Sample URL: http://<PSC-HOST>/RequestCenter.

Step 3 From the Administration module in the module selection drop down, click **Directories**.

Step 4 Click Add to add a new data source.

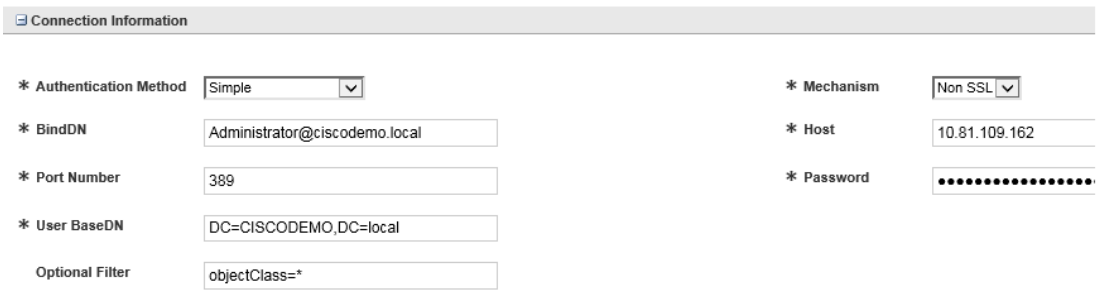
Step 5 Enter a name for Data Source Name.

Step 6 Select LDAP for the Protocol.

Step 7 Select MS Active Directory for Server Product.

Step 8 Set the connection information, as shown in Figure 4-1.

Figure 4-1 AD Connection

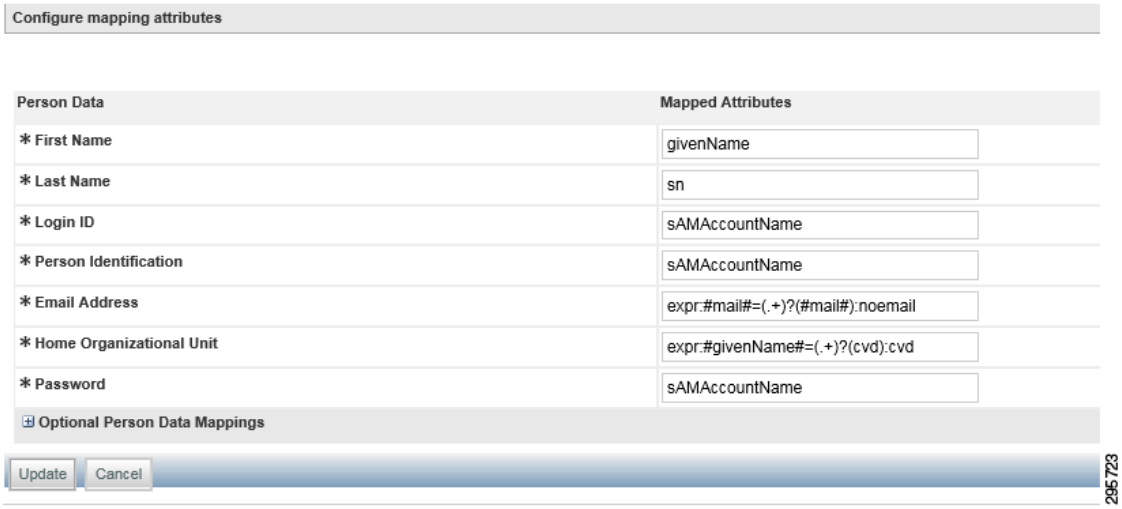


Step 9 Click the Mappings tab (right side).

Step 10 Enter a mapping name.

Step 11 Configure mapping attributes, as shown in Figure 4-2.

Figure 4-2 Configure Mapping Attributes

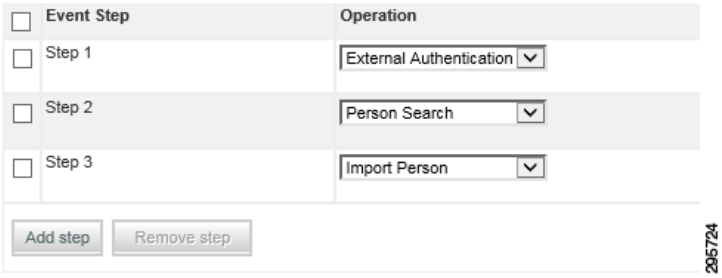


Step 12 Click the Events tab (right side).

Step 13 Click Edit at Login event.

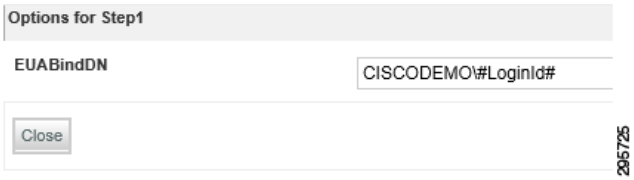
Step 14 Add the steps as shown in Figure 4-3.

Figure 4-3 Add Event Steps



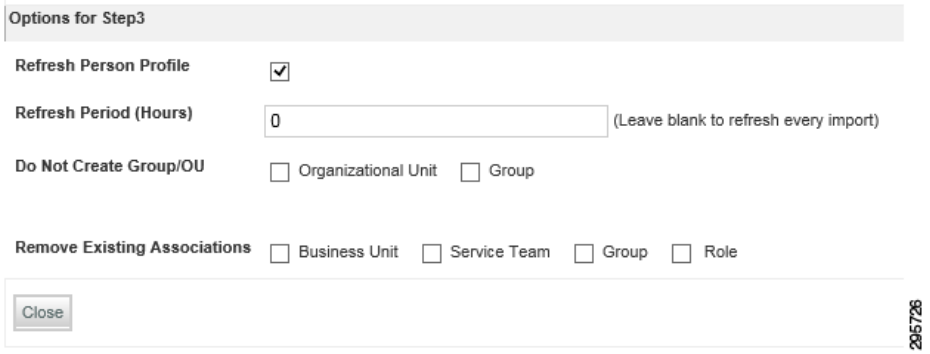
a. For Step 1, click Options and set the value as shown in Figure 4-4.

Figure 4-4 Set Values for Step 1



b. For Step 3, click Options and set the value as shown in Figure 4-5.

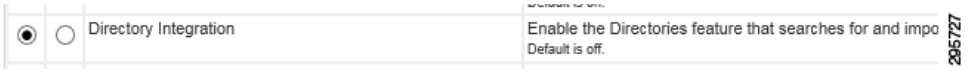
Figure 4-5 Set Values for Step 3



Step 15 Click the Settings tab (on the top).

Step 16 Turn on directory integration by selecting the radio button, as shown in Figure 4-6, and then click Update.

Figure 4-6 Turn on Directory Integration

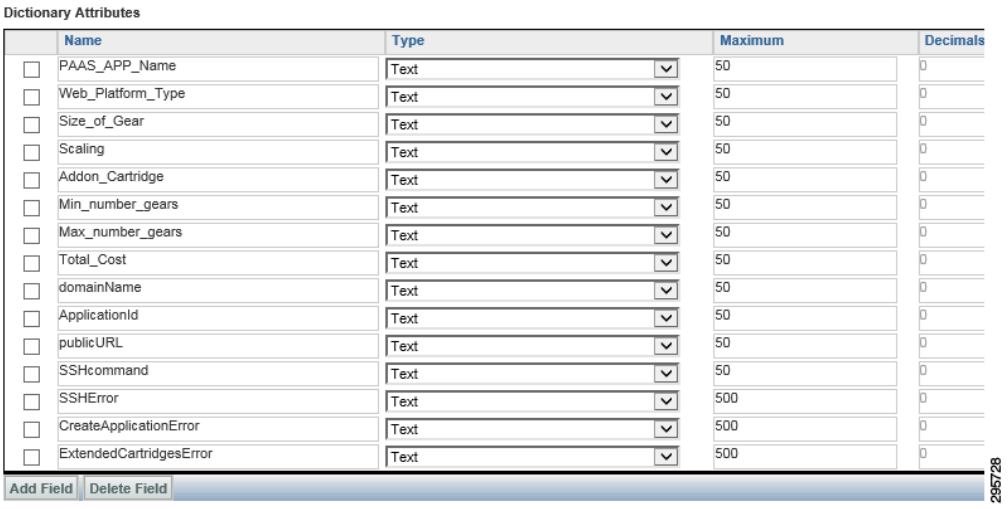


Creating a New "PAAS Application" Service

Perform the following procedure to create a new PAAS application service.

Step 1 Create a dictionary using the interface shown in Figure 4-7.

Figure 4-7 Create a Dictionary



Step 2 Create the necessary Javascripts as shown in Figure 4-8, Figure 4-9, and Figure 4-10.

Figure 4-8 Create the Necessary Javascripts 1

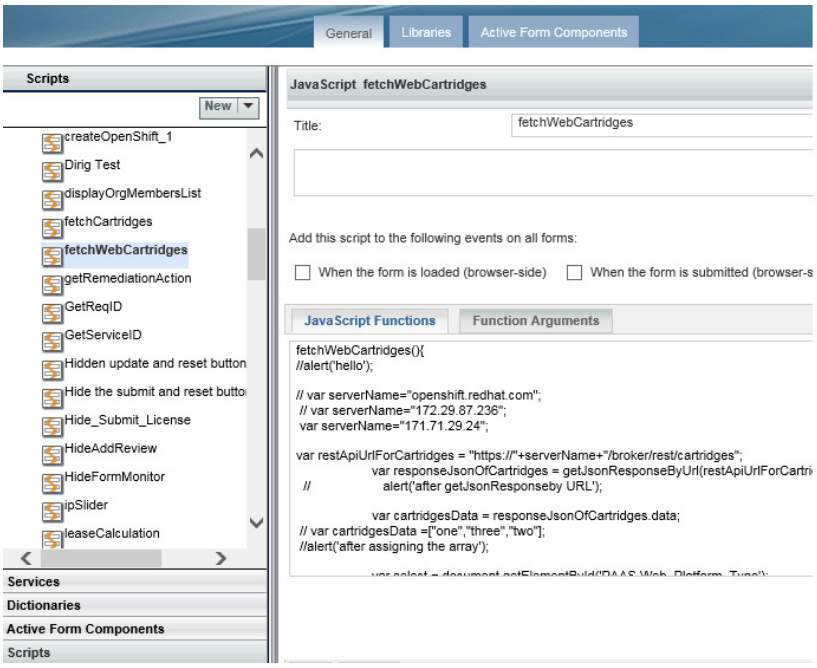


Figure 4-9 Create the Necessary Javascripts 2

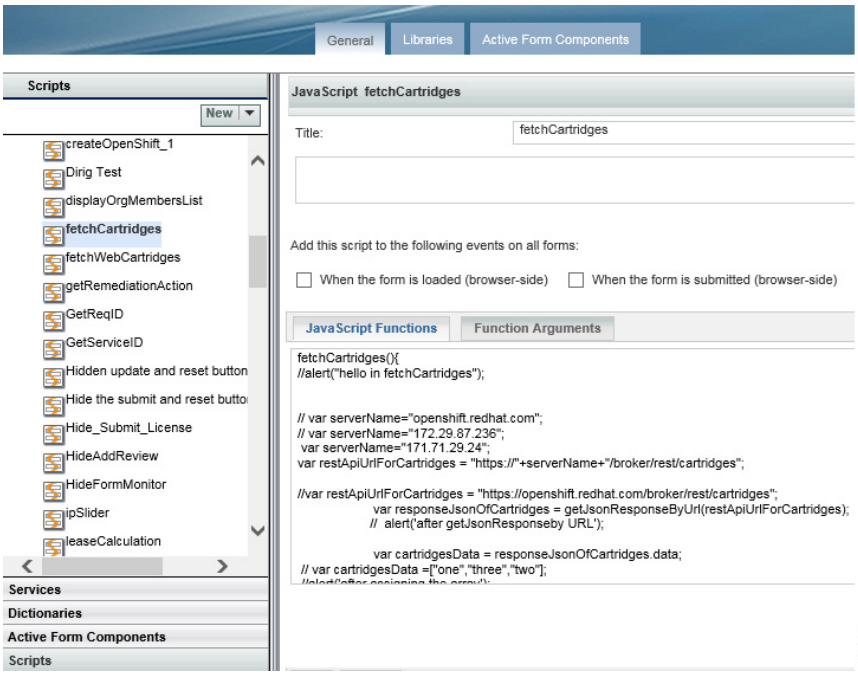
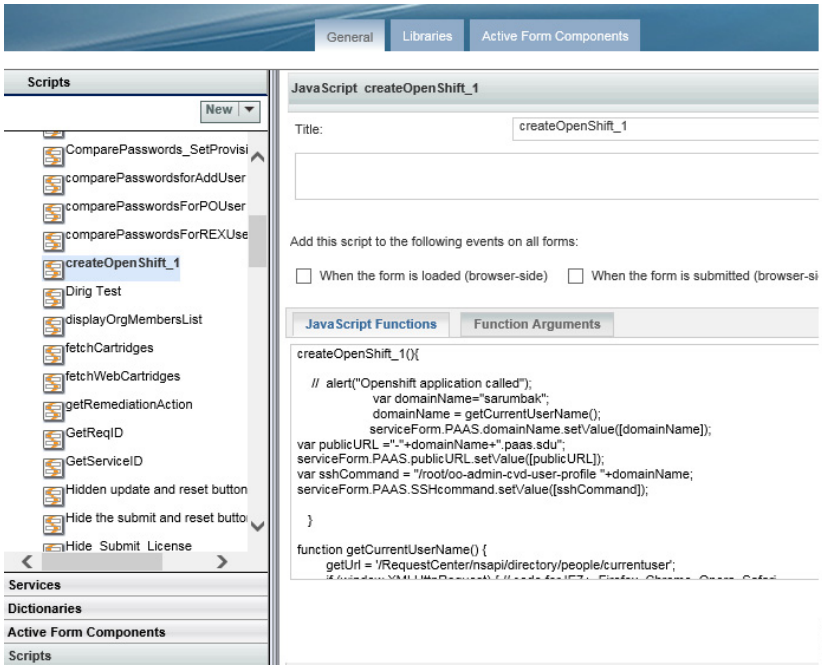


Figure 4-10 Create the Javascripts Needed 3



Creating an Active Form Component

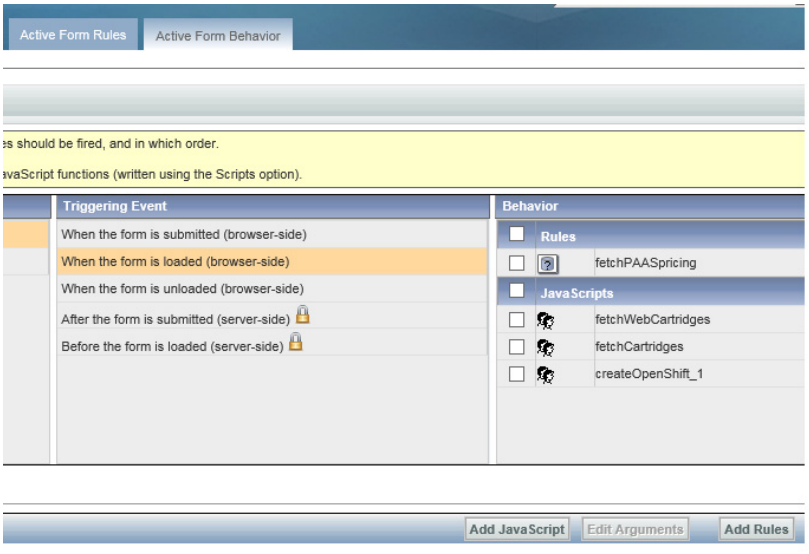
Perform the following procedure to create an active form component.

Step 1 Add the dictionary that was created in the first step.

Step 2 Set the display properties of each field.

Step 3 Set the Active Form behavior; add the scripts (that were created in the previous step), as shown in Figure 4-11:

Figure 4-11 Add Scripts

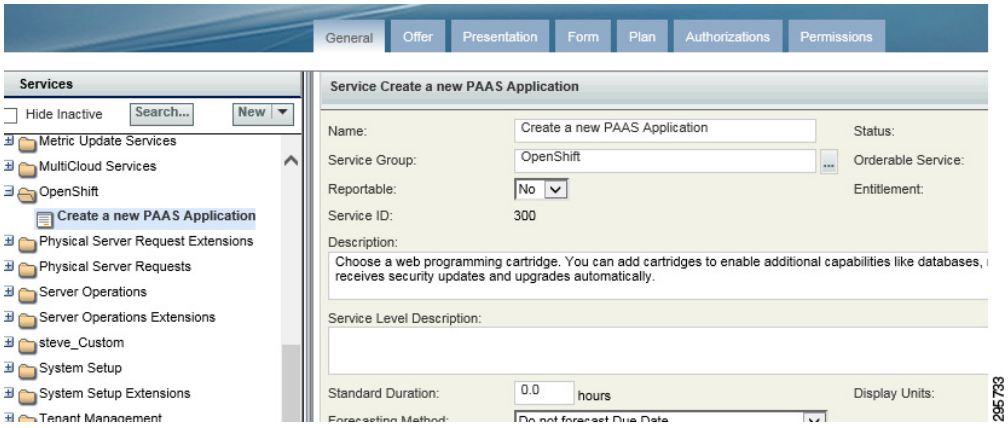


Creating a Service

Perform the following procedure to create a service.

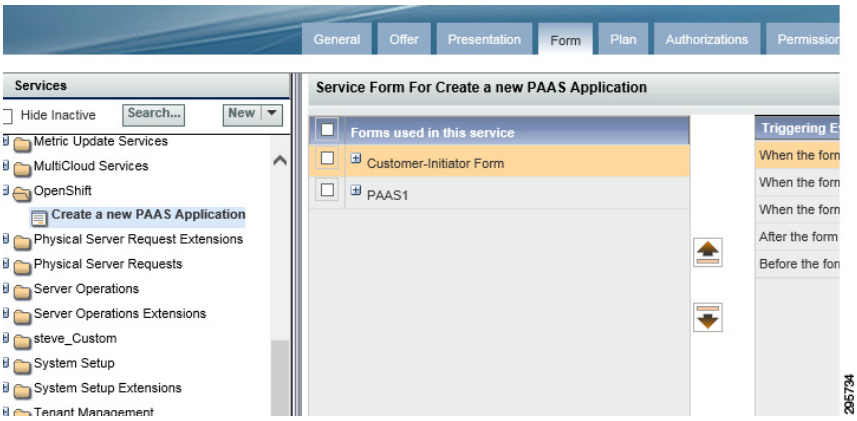
Step 1 Create a service with desired name and set the description as shown in Figure 4-12.

Figure 4-12 Create a Service



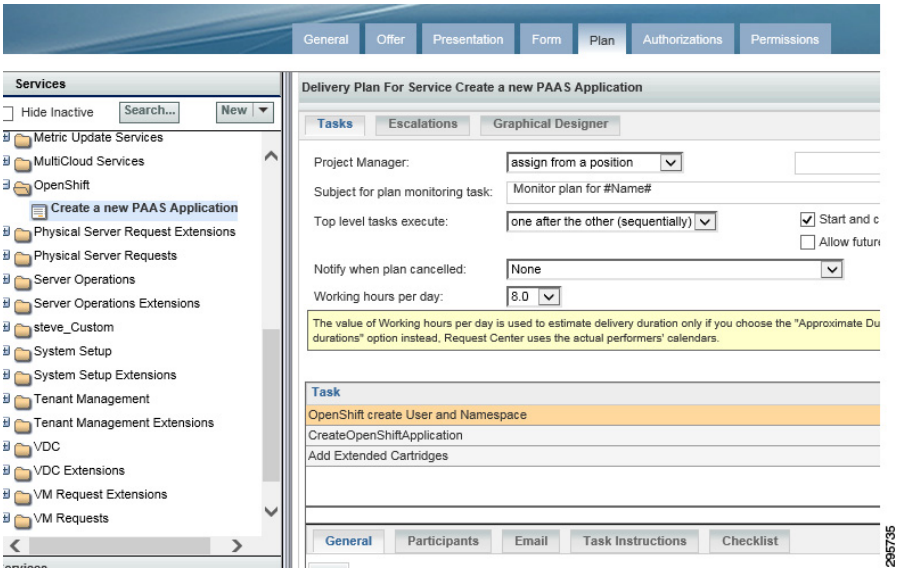
Step 2 Add the Active form component that was created to this service, as shown in Figure 4-13.

Figure 4-13 Add Active Form Component



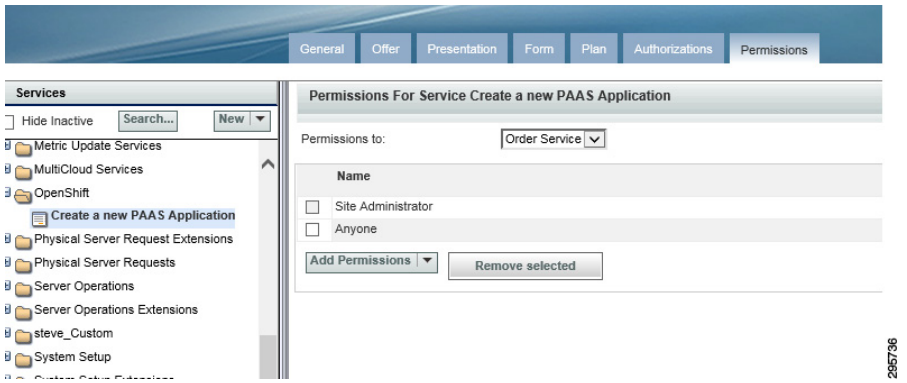
Step 3 Set the delivery plan, as shown in Figure 4-14.

Figure 4-14 Set Delivery Plan



Step 4 Set permissions, as shown in Figure 4-15.

Figure 4-15 Set Permissions



ServiceLink Agent Configuration

Three ServiceLink agents need to be configured.

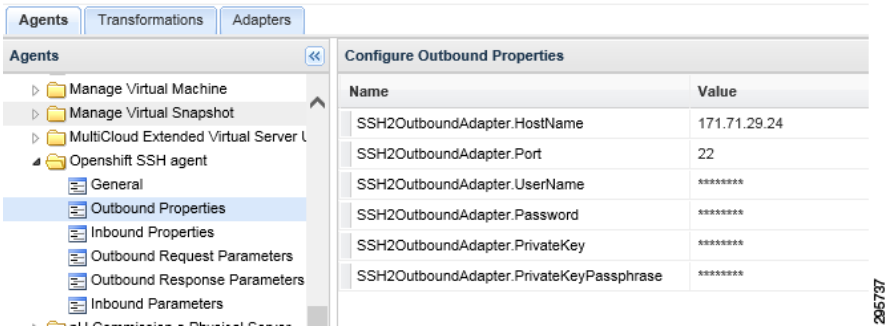
- OpenShift SSH Agent, page 4-24
- Creating an OpenShift Application Agent, page 4-25
- Add Extended Cartridges Agent, page 4-27

OpenShift SSH Agent

Perform the following procedure to configure OpenShift SSH Agent.

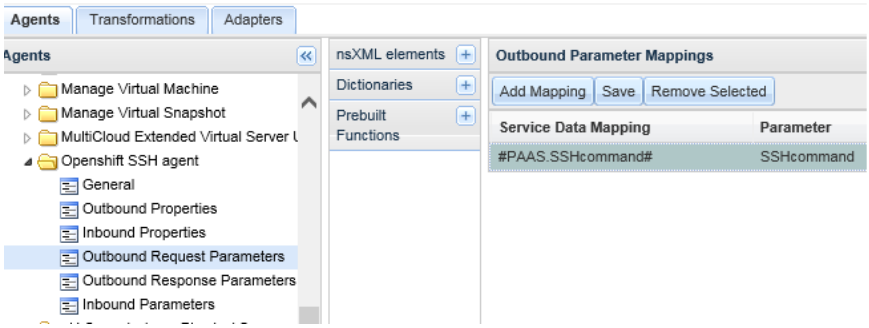
Step 1 Configure ServiceLink agent Outbound Properties, as shown in Figure 4-16.

Figure 4-16 Configure ServiceLink Agent Outbound Properties



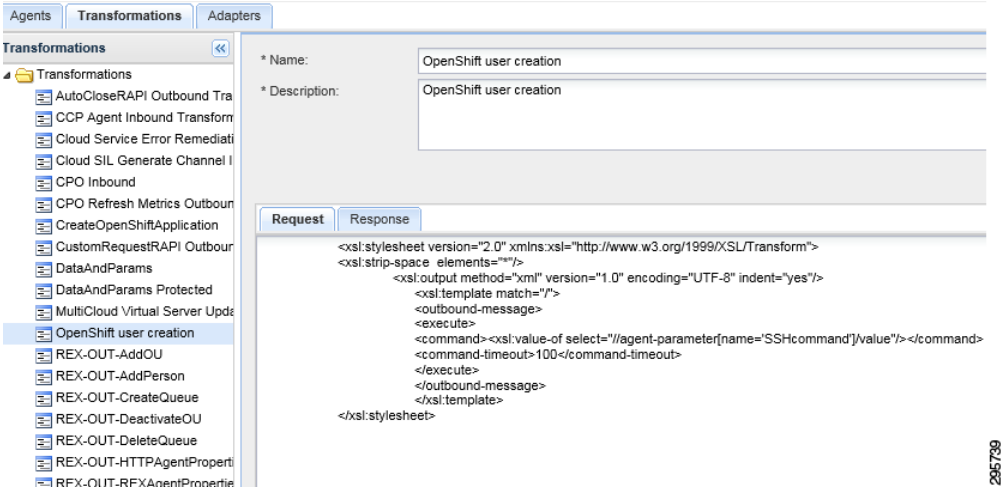
Step 2 Configure Outbound Request parameters, as shown in Figure 4-17.

Figure 4-17 Configure Outbound Request Parameters



Step 3 Configure Outbound Transformation request, as shown in Figure 4-18.

Figure 4-18 Configure Outbound Transformation Request

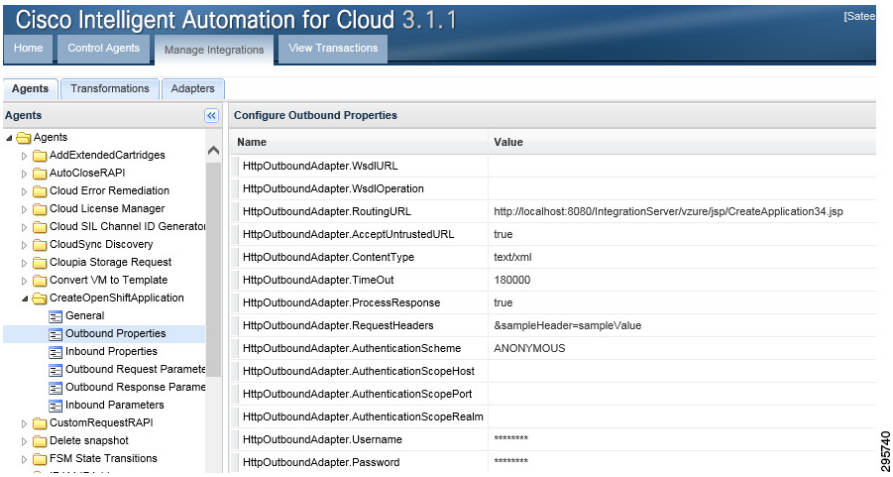


Creating an OpenShift Application Agent

Perform the following procedure to create an OpenShift application agent.

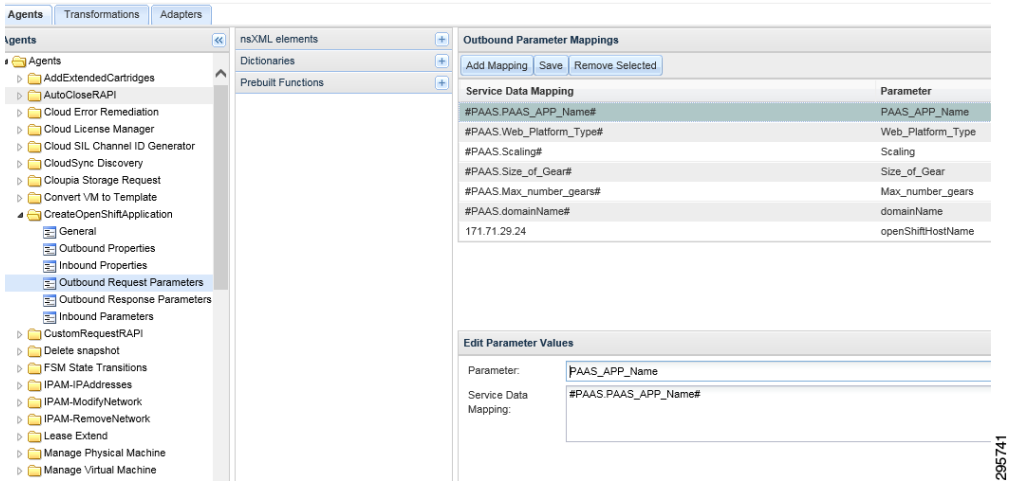
Step 1 Configure ServiceLink agent Outbound Properties, as shown in Figure 4-19.

Figure 4-19 Configure ServiceLink Agent Outbound Properties



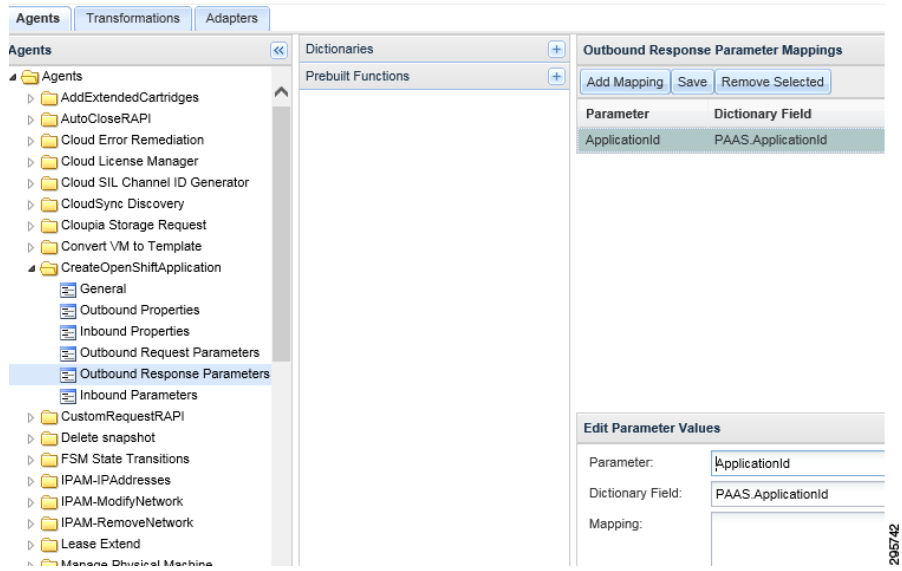
Step 2 Configure Outbound Request parameters, as shown in Figure 4-20.

Figure 4-20 Configure Outbound Request Parameters



Step 3 Configure Outbound Response Parameters, as shown in Figure 4-21.

Figure 4-21 Configure Outbound Response Parameters

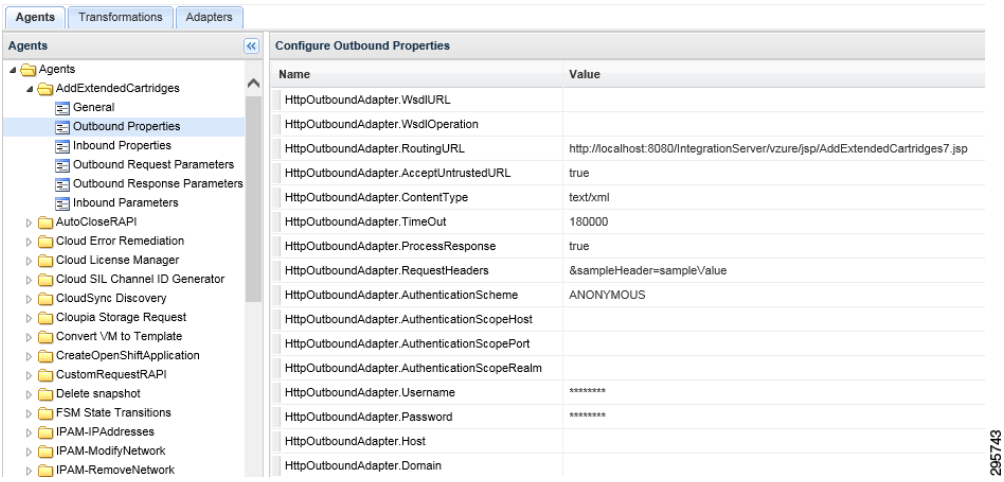


Add Extended Cartridges Agent

Perform the following procedure to add extended cartridges agent.

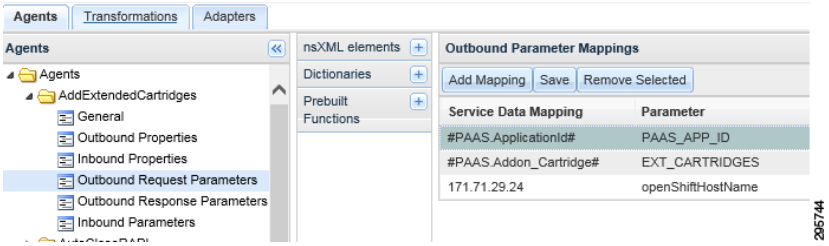
Step 1 Configure ServiceLink agent Outbound Properties, as shown in Figure 4-22.

Figure 4-22 Configure ServiceLink Agent Outbound Properties



Step 2 Configure Outbound Request parameters, as shown in Figure 4-23.

Figure 4-23 Configure Outbound Request Parameters



Creating Portal Pages

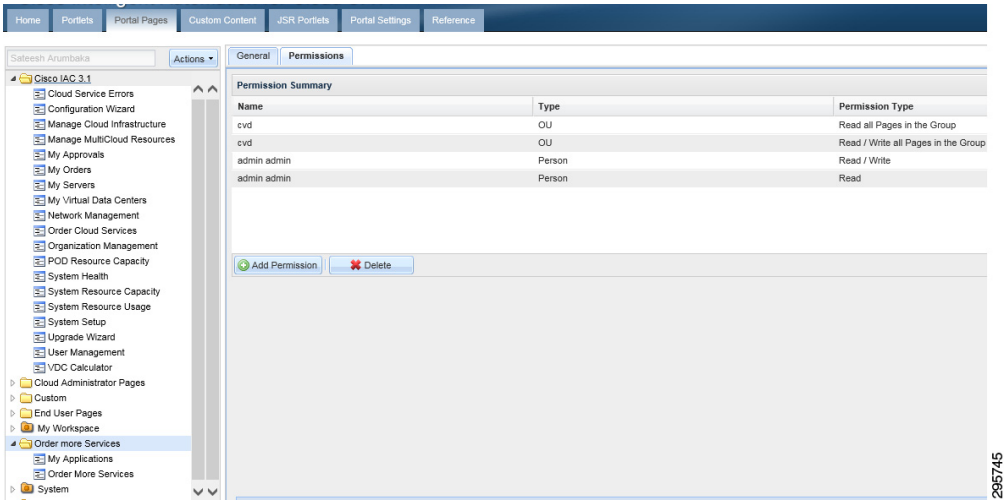
Perform the following procedure to create portal pages.

Step 1 Log in to Portal Designer module and create a new Portal Page Group.

Step 2 Set the permissions so that the Organization Unit has read and write permissions to all the pages in the Portal Page group.

Step 3 Create two Portal pages in this group (Figure 4-24).

Figure 4-24 Create Portal Pages



Step 4 Create three portlets, as shown in Figure 4-25, Figure 4-26, and Figure 4-27.

Figure 4-25 Create Portlets 1

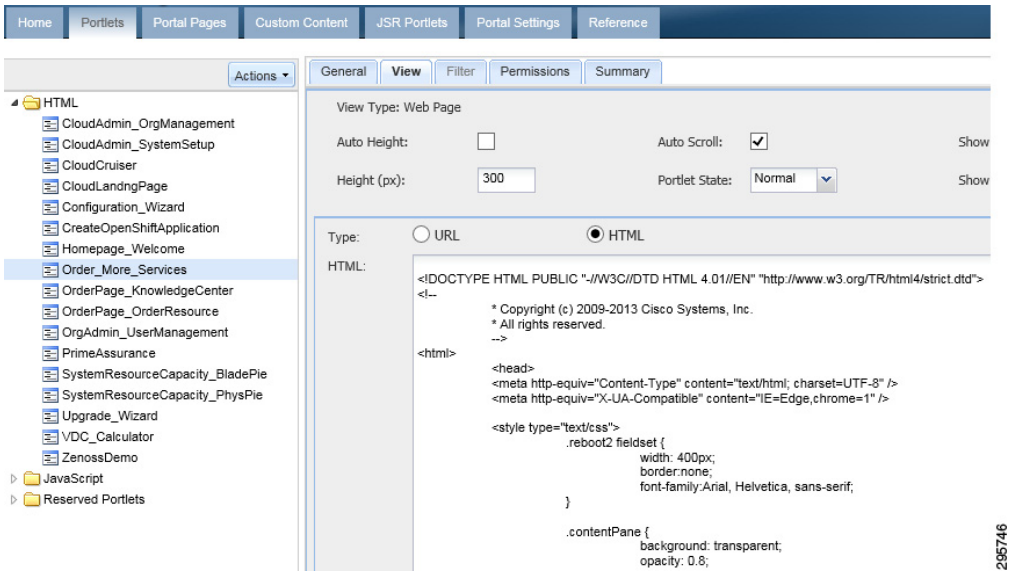


Figure 4-26 Create Portlets 2

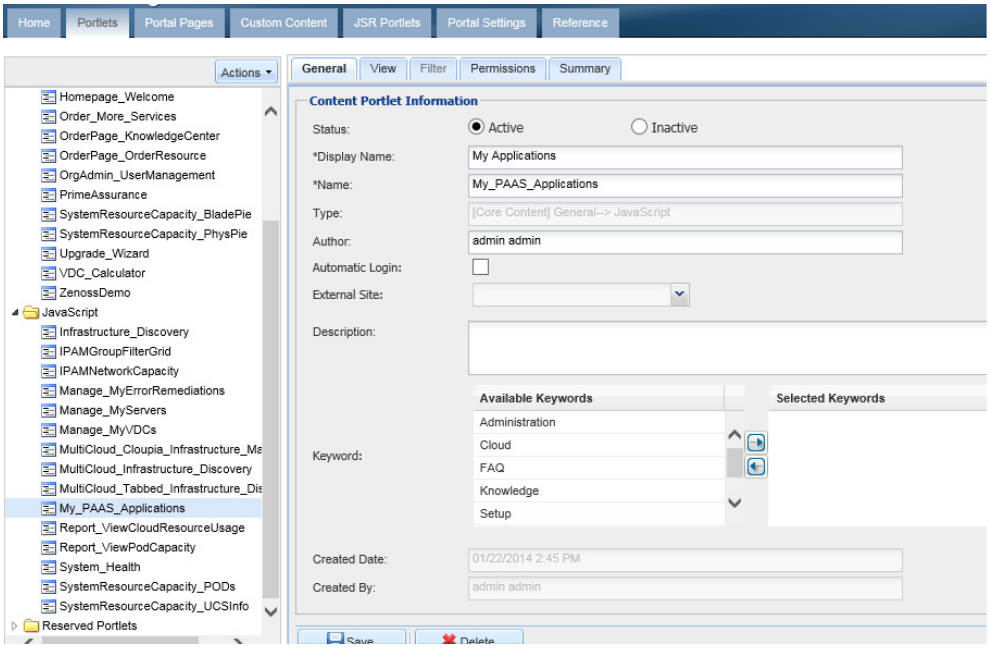
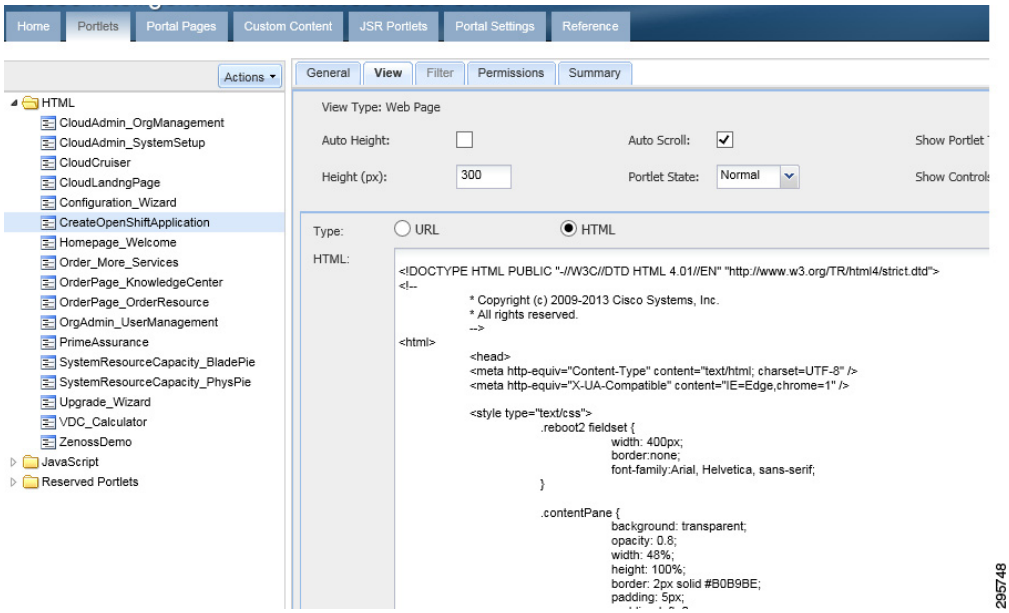


Figure 4-27 Create Portlets 3



Step 5 Set the permissions on all three of the Portlets so that the Organization Unit has read and write permissions. Add these portlets to the portal pages, as shown in Figure 4-28 and Figure 4-29.

Figure 4-28 Add Portlets to Portal Pages 1

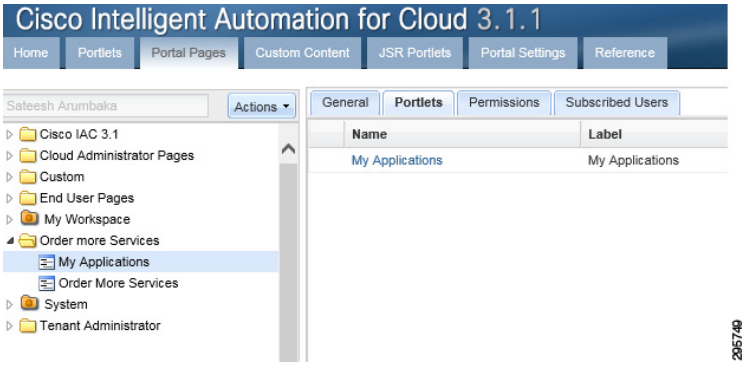
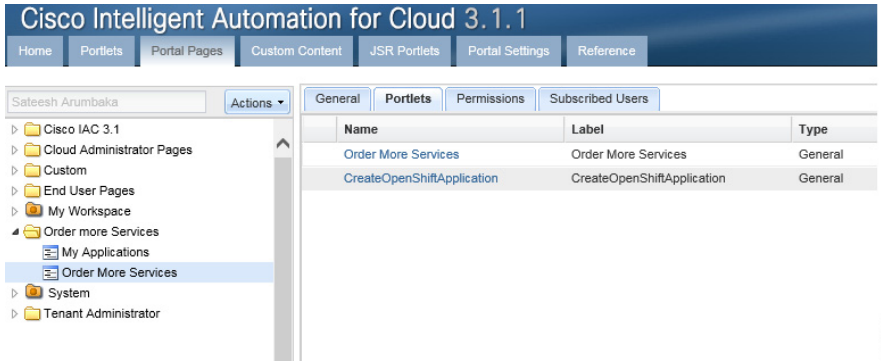


Figure 4-29 Add Portlets to Portal Pages 2



Deploy Java Server Page (JSP)

Perform the following procedure to deploy a Java server page.

Step 1 Log on to the Cisco Prime Catalog server.

Step 2 Deploy the attached JSPs to the following locations:

- C:\jboss-as-7.1.1.Final\standalone\deployments\RequestCenter.war\vzure\jsp
- C:\jboss-as-7.1.1.Final\standalone\deployments\ServiceLink.war\vzure\jsp

OpenShift Broker admin Script

Perform the following procedure to execute an OpenShift admin script.

Step 1 Log on to the Broker host.

Add osadmin to the openshift broker as a user

```
#login to host
ssh -i <sshkey> root@<brokerhost>
#create osadmin user
    oo-admin-ctl-user -c -l osadmin
```

Step 2 Change osadmin user to allow a large number of domains and gears. All domains are going to be owned by osadmin. Domains for users are assigned to osadmin, and the user is granted edit access to the domains. This allows the user to create/delete/manage applications in their domain.

```
#set max domains and max gears
oo-admin-ctl-user -l osadmin --setmaxdomains 10000
    oo-admin-ctl-user -l osadmin --setmaxgears 10000
```

Step 3 Add the attaché oo-admin-user-profile profile creation scripts into the broker hosts—this is required for PSC to log into the broker host to create openshift users and assign them domains. This script does the following:

1. Creates the user logins.

2. Creates the domain.
3. Assigns the domain to osadmin.
4. Assigns the user edit access to the domain.
5. Limits the max number of gears for the user to 0.
6. Limits the max number of domains for the user to 0.

```
# copy the script to openshift broker host
scp oo-admin-cvd-user-profile root@<brokerhost>
# edit the oo-admin-cvd-user-profile file to set the following parameters
# OSADMIN="osadmin" # set this to the osadmin user as shown
# OSPWD="<password>" # set this to the osadmin directory/ldap password
    # BROKER="<broker host ip address>" # set this to the public ip of the broker
```

Testing

The methodology used for testing and validating the functionality was based on test cases per use case for the use cases outlined in the System Overview section.

Use Case 1: Integrated Provisioning for IaaS and PaaS

The following procedure is performed for configuring integrated provisioning for IaaS and PaaS.

Step 1 Add users to the Active Directory installation in the specified OU of the implementation section.

Step 2 Log in to PSC.

Step 3 From the Portal selection dropdown, choose My Workspace.

Step 4 From the workspace, click the Order more services tab.

Step 5 On the Platform As A Service Portlet, click Create a new PAAS Application.

Results

The following results are observed.

- Able to log in with created credentials.
- Able to view Unified portal page with IaaS Cloud Services and PaaS Services.
- Have permissions to order PaaS Application, which shows up, and order form..

Use Case 2: Application Stack Creation

The following procedure is performed for configuring application stack creation.

Step 1 Verify the order form shows up after step 5 in Use Case 1: Integrated Provisioning for IaaS and PaaS, page 4-32.

Step 2 Enter the application name that you want to create, e.g. rubyapp.

Step 3 Choose platform type (eg.g python-2.7).

Step 4 Choose Gear Profile Small.

Step 5 Click False on the Scaling type.

Step 6 Click Submit Order.

Step 7 Close the order confirmation form.

Results

The following results are observed.

- Able to choose different application stacks.
- Able to choose different resource profiles.
- Able to submit order for stack creation.
- Able to confirm order for stack creation.

Use Case 3: Single Pane Management of Application Stack

The following procedure is performed for configuring single pane management of application stack.

Step 1 On the workspace page, click the My Applications tab.

Step 2 On the list of applications, click the application name just created in Use Case 2: Application Stack Creation, page 4-32.

Step 3 In the Take Action section:

a. Click View App to take you to the application URL.

b. Click Stop application.

c. Click Start application.

Step 4 For scaled application type, click View Status.

Step 5 Use git clone to clone the application repository shown in the Application Details section.

Step 6 In the Take Action section, click Delete application.

Step 7 Repeat Use Case 2: Application Stack Creation, page 4-32 steps, click Scaled Application, and submit order.

Step 8 Repeat Use Case 2: Application Stack Creation, page 4-32 steps and create a few different applications with different application stacks.

Step 9 On the list of applications, verify Steps 3 through 7 for each application.

Results

The following results are observed.

- Able to view list of applications created.
- Able to act on different actions like start, stop for each application stack.
- Able to delete application.
- Able to browse to application URL.

Use Case 4: Integrated Provisioning for IaaS and PaaS

The following procedure is performed for configuring integrated provisioning for IaaS and PaaS.

Step 1 Refer to Use Case 1: Integrated Provisioning for IaaS and PaaS, page 4-32 and Use Case 2: Application Stack Creation, page 4-32, where a direct order for both IaaS and PaaS resources can be placed from a single pane. Validate this use case.

Note Ordering of IaaS resources comes with the PSC solution and is not validated separately.

Results

See results for Use Cases 1, 2, and 3.

Use Case 5: Network-based Segmentation of PaaS Districts for Security

The following procedure is performed for configuring network-based segmentation of PaaS districts for security.

Step 1 Repeat steps 1, 2 and 3 in Use Case 2: Application Stack Creation, page 4-32.

Step 2 Choose Small-Secure* Gear profile.

Step 3 Submit order.

Step 4 Repeat steps 1 through 8 in Use Case 3: Single Pane Management of Application Stack, page 4-33 for the secure network application.

Step 5 Log in to OpenShift node in secure network as admin/root via ssh and verify that application was created.

Results

The following results are observed.

- Able to choose a secure network profile for an application.
- Able to verify that the application got created in secure network.

Summary

The following recommended implementation was conducted:

- Use of OpenStack for IaaS.
- Deployment of OpenShift Enterprise into OpenStack using HEAT templates.
- Deployment of two separate network segments in OpenStack to house secure and standard OpenShift nodes.
- Creation of PaaS ordering Services that talk to OpenShift Broker in PSC.
- Creation of PaaS Application management portal pages in Prime Services Catalog.
- Binding OpenShift and OpenStack authentication to LDAP (AD) directory.