Redhat Openshift installation on the SysEleven Openstack Cloud

At SysEleven we have been operating Openstack public clouds since 2016. We currently run five datacenters across Germany with public Openstack resources. We offer our customers managed and self managed resources. These include virtualized machines and Ceph storage. IPv4 and IPv6 networking and Load balancers. Customers can use our API endpoints to provision their infrastructure via our extensive management API or with our provided terraform providers. We also offer a managed Kubernetes product called MetaKube. We currently host and run over 400 Kubernetes clusters for customers and internal usage. I installed and configured a Redhat Openshift Cluster in our datacenter in Hamburg to run the provided tests with the OpenShift Provider Compatibility Tool. I performed the following steps to install and configure Openshift on the SysEleven Openstack Cloud:

1. I created a terraform module to provision the infrastructure as described in the documentation for the OpenShift Provider Compatibility Tool. I deployed a total of six servers. Three of the servers were used for the control plane. The other three servers were configured as compute nodes (worker nodes). Each virtual server was configured with the flavor m2.medium. Which has four virtual CPU's and 16GB of RAM:

Name	Openstack Flavor	Networks
openshift-control1	m2.medium	openshift-nodes-net=10.129.14.75,109.68.228.200
openshift-control2	m2.medium	openshift-nodes-net=10.129.14.20,109.68.228.240
openshift-control3	m2.medium	openshift-nodes-net=10.129.14.55,109.68.228.78
openshift-worker1	m2.medium	openshift-nodes-net=10.129.14.172,109.68.229.249
openshift-worker2	m2.medium	openshift-nodes-net=10.129.14.181,109.68.228.178
openshift-worker3	m2.medium	openshift-nodes-net=10.129.14.55,109.68.228.78

Our regular servers have 50GB of ephemeral storage. If more storage is necessary we advise customers to add additional block storage with volumes. To fulfill the requirements I needed to add an extra 100GB volume to each server. I also added a 5GB bootable volume to attach the install ISO which Redhat provided via the assisted installer which I used to install the Openshift

Cluster. I also created three individual Cloud Load Balancers based on Openstack Octavia. I pointed three DNS A-Records to the created cloud Load Balancers:

Name	Description	DNS Records
openshift-lb-api	API LB	api.syseleven.example.com
openshift-lb-ingress	ingress controller LB	*.apps.syseleven.example.com
openshift-lb-internal	internal LB / ignition	api-int.syseleven.example.com

2. I used the Redhat Openshift assisted installer to perform the installation via the Redhat Hybrid Cloud Console. I used the Infrastructure provider Bare Metal x86_64. I added a cluster and a domain name for the cluster. I did not modify any other options or install any operators on the next page.
3. The bootable ISO image was generated and downloaded to my client machine. Then I uploaded the ISO image to Openstack Glance via the API (image Store) to boot the Openshift VM's from.
4. I used the Terrform module I created earlier to provision the necessary Openstack infrastructure to create the cluster resources needed. All virtual machines were booted with the ISO image provided by the Redhat Openshift installer. After a few minutes all of the machines were visible in the assisted installer. Then I configured the roles control plane node and compute node.
5. Configuring the networking was the next step. I chose the user-managed network option. I provided the Machine network, the API Server virtual IP and the Ingress virtual IP. I also selected the advanced networking options. Where I defined the Cluster network CIDR the Cluster network host prefix and the Service network CIDR. I also added the A-Records pointing to the cloud Load Balancers described above to our DNS service.
6. The last step in the assisted installer was to perform the installation itself. The machines are bootstraped. The control plane nodes start to form the cluster and the compute nodes join the cluster. All nodes are rebooted before the installation is complete.
7. I downloaded the kubeconfig from the assisted installed page and connected to the Openshift API and the Webui. There were some outstanding Openshift updates which needed to be installed before I could continue.
8. Depending on your CNI configuration if you are using IP encapsulation or not. You may need to add the allowed address pairs of your POD CIDR to the Openstack VM ports which are connected to the cluster nodes. The openstack cli command should like similar to this:

   openstack port set --allowed-address ip-address=172.16.0.0/12 $PORT-UUID

9. I added the terraform module I created to GitHub in case you would like to review it. You can find it here:

https://github.com/syseleven/openshift-terraform