Red Hat Consulting

Crédit Agricole

OpenShift Container Platform 4.14

Documento di installazione

Ambiente di Collaudo

Red Hat Consulting

Confidentiality, Copyright, and Disclaimer

This is a Customer-facing document between Red Hat, Inc. and Crédit Agricole.

This document is not a quote and does not include any binding commitments by Red Hat.

Trademarks

Trademarked names may appear throughout this document. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, the names are used only for editorial purposes and to the benefit of the trademark owner with no intention of infringing upon that trademark.

Review History

Version	Date	Contributor	Role	Description
1.0	2023-11-28	Giulia Bacchini	Senior Cloud Consultant	Ambiente di Collaudo

Red Hat Consulting

Table of Contents

1. Introduzione

Red Hat è stata ingaggiata da Crédit Agricole per installare un cluster Openshift su piattaforma vSphere.

Questo documento tratta l'installazione del cluster, denominato cluster "di collaudo", ospitato su piattaforma vSphere. La modalità di installazione scelta per questo ambiente è la "IPI", la versione di OpenShift Container Platform scelta è stata la 4.14.2.

1.1. Purpose

Questo documento descrive la procedura di installazione del cluster e dei vari servizi compresi.

1.3. Termini e acronimi

La tabella di seguito indica il significato di alcuni termini e acronimi utilizzati nel documento.

Acronym	Description
RH	Red Hat, Inc
RHEL	Red Hat Enterprise Linux
AD	Active Directory
CA	Certificate Authority
DC	Data Centre
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
FQDN	Fully Qualified Domain Name
Guest	Also see "VM". This is virtual machine running on a Host.
HA	High-Availability or Highly-Available
Host	The physical hardware or the logical OS which runs virtualisation technology allowing one or more Guest OS's to run on the hardware owned by the Host
Compute Nodes	Compute nodes dedicated to host end user containers and apps
Infrastructure Nodes	Compute nodes reserved to host infrastructure services like routing layer/metrics/logging
Master Node	Node acting as controller for OCP, exposing api and hosting cluster configuration
IPI	Installer Provided Infrastructure

OS	Operating System
OCP	OpenShift Container Platform
SAN	Storage Area Network
SSL	Secure Sockets Layer
VIP	Virtual IP address
VLAN	Virtual LAN is a networking virtualisation technology
VM	Virtual machine, in OSP terms, synonymous with “Workload” or “Guest”
VXLAN	Virtual Extensible LAN (VXLAN) is a network virtualisation technology
Workload	Synonym for “Guest” or “VM”
RHOSP	Red Hat OpenStack Platform
OCP	Openshift Container Platform
ODF	Openshift Data Foundation
RHACM	Red Hat Advanced Cluster Management
AZ	Availability Zone
HA	High Availability

2. Installazione minimale

Configurazioni preliminari

Download Installer

Per il cluster di collaudo è stata scelta la versione di Openshift Container Platform 4.14.2.

Per installare questa specifica versione, è stato scaricato l'installer, il client e la pull-secret dal portale https://console.redhat.com/openshift/install, navigando la sezione "Datacenter", selezionando la piattaforma vSphere, e la modalità *full-automated*.

Il tutto è stato posizionato sul bastion all'interno della directory /roo/ocp4-free-collaudo/.

A questo punto sono stati estratti installer e client con i seguenti comandi:

```
Unset
# cd /roo/ocp4-free-collaudo/
# tar zxvf openshift-install-linux.tar.gz
# chmod +x openshift-install
# tar zxvf openshift-client-linux.tar.gz -C /usr/bin
# chmod +x /usr/bin/oc
# oc completion bash >/etc/bash_completion.d/openshift
```

Configurazione chiave ssh

Per creare la chiave ssh, così da permettere la login ai nodi una volta installati, sono stati eseguiti i seguenti comandi:

```
Unset
# ssh-keygen -t ed25519 -N '' -f  /root/.ssh/ocp4key
# eval "$(ssh-agent -s)"
# ssh-add /root/.ssh/ocp4key
```

Certificati vCenter

Per permettere all'installer di comunicare con le API del vCenter via https, è necessario scaricare sul nodo *bastion* i certificati del vCenter.

Per fare questo è possibile sia collegarsi direttamente alla dashboard del vCenter e scaricarli dalla sezione "**Download trusted root CA certificates**", che eseguire i seguenti comandi:

```
Unset
# cd /roo/ocp4-free-collaudo/
# curl -k -O https://AC-CAGS-VCSA001.cariprpc.it/certs/download.zip
# unzip download.zip
# cp certs/lin/* /etc/pki/ca-trust/source/anchors
# update-ca-trust extract
```

Creazione cluster

Per installare il cluster è stato preparato il seguente install-config.yaml file:

```
Unset
apiVersion: v1
baseDomain: cariprpccoll.it
proxy:
 httpProxy: http://vip-navproxy-server.cariprpc.it:8080
 httpsProxy: http://vip-navproxy-server.cariprpc.it:8080
 noProxy:
localhost,127.0.0.1,localaddress,.localdomain.com,.cariprpccoll.it,10.68.0.0/14,172
.27.0.0/16,.cariprpc.it,10.215.86.0/24
compute:
- name: worker
 hyperthreading: Enabled
 platform:
  vsphere:
   cpus: 8
   coresPerSocket: 2
   memoryMB: 32768
   osDisk:
    diskSizeGB: 120
 replicas: 6
controlPlane:
 hyperthreading: Enabled
 name: master
```

```
 platform:
  vsphere:
   cpus: 8
   coresPerSocket: 2
   memoryMB: 32768
   osDisk:
    diskSizeGB: 120
 replicas: 3
metadata:
 name: ocp-collaudo
platform:
 vsphere:
  apiVIP: 10.215.86.4
  cluster: "OCP_PREPROD"
  datacenter: "ACILIA"
  defaultDatastore: ESX-OCP-PREPROD-AC-0000
  ingressVIP: 10.215.86.3
  network: "dvpg_619_DMZ_ocprhel_col"
  password: 'N65sCPnrD$AasJANJThc'
  username: cariprpc\cp_12_vcenter_OCP
  vCenter: ac-cags-vcsa001.cariprpc.it
  folder: "/ACILIA/vm/COLLAUDO/OCP"
publish: External
pullSecret:
'{"auths":{"cloud.openshift.com":{"auth":"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc
3NfYTg3ZmEzMjYwMTRmNDI5M2I3MjYxZWFmMGE0NzVhOTY6OVBDU0k0S1RVNlFQNlM4T09JSDJINDNQRDQy
Rlk5MlkwN0pYUzVVTzk1UTk4RkxFMExVNlpUWExOUDVMQjI4NQ==","email":"cesarezuppa@cagroups
olutions.it"},"quay.io":{"auth":"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfYTg3Zm
EzMjYwMTRmNDI5M2I3MjYxZWFmMGE0NzVhOTY6OVBDU0k0S1RVNlFQNlM4T09JSDJINDNQRDQyRlk5MlkwN
0pYUzVVTzk1UTk4RkxFMExVNlpUWExOUDVMQjI4NQ==","email":"cesarezuppa@cagroupsolutions.
it"},"registry.connect.redhat.com":{"auth":"fHVoYy1wb29sLTA0Yzk4ZmQ1LTZkMTUtNDY3NC1
hYzc5LWQ3ZWZlZGY5ZDc4YzpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSmpaVFprWVRZMlpXVT
NOekUwWkdGallUUmhORE14TUdJM1pUTXhZVGszTXlKOS5Vc0xKMHlOTG9WNjdMdTFnOGxwMjdmRXRsdVNJW
XQ2WVE2endQSTlZVWxCUFNGQW9ETWlfTjhvSGNxcXQ3VTFUdUpmMnlvUUtyVHdvbkxucG9kdVBxMktGV2RD
MHVON2l1VTQ4V3RyOTF1VDdGM0ZoT0w2c3c5bTR1RW42OGhNTFVPYWxSR3BkWF9feGZUeG5jc1lOSXZqSG5
kLUFreWJCNlJiRzJKRm1DLXlzZW1XVThpbk9aV09Qc21xcEJWOVloSGlXZFR2d05zeVlRNW05b1VvUy1qWU
VWLXpxZ0NtTFpsczdaT2RKRE9oOTR5UGZUZm1XSV9HYnh5S05ydS05aHhkZE5MTGRXaXBCRmVreTluX1Bua
nZtaFo2bnRRaUtxRXJXMURPLTJmdDEtcW9tLUlxcWRGZGZpM3FzSXVlQ1BEM1puNDJ2Nm5URFk4ZXRZOVdm
cUxra2JSdjl6eXRPMThrM285clNwSlh1dkZyU01lVmNKZTNPcmJHOTFpYmdfUF9XNU5pX1hZVndKOG0xT3F
ZLTJoQUN6U3ltTXRXZWJkZkdRUmZCVi1heElMYi1NYktHdjBjcUI2VEhsOXVhQ0ptTXVvRkFkUFZmeV8yLT
lWMjdpbXRsLXJSd0lkbVNHWWgzUTN3aG44U196cnc3U2xpRW1VSDRUdzJpVTg5SWRWd1FndXd4dEcxVjhMb
3VfMW1HVmt4XzhpTTFLSDhOeDNMOFBiT2JiZXBpZWJDYWVuamt1QnFWQ1dubl9pS0hrSXVQTHZTQ3B2am9P
SUhNcTc2WDZFUGtxU29oYl9QR2NhaVJNTTVHTkVxdHZVaVJ0RDNhRzZDVm1SSFk5VEQ1ZDFGQjJCZ01xYjR
ONGUza1pLejFSbzY4dG5BX0pBSV9teFpRaU5fSWdHdw==","email":"cesarezuppa@cagroupsolution
s.it"},"registry.redhat.io":{"auth":"fHVoYy1wb29sLTA0Yzk4ZmQ1LTZkMTUtNDY3NC1hYzc5LW
Q3ZWZlZGY5ZDc4YzpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSmpaVFprWVRZMlpXVTNOekUwW
kdGallUUmhORE14TUdJM1pUTXhZVGszTXlKOS5Vc0xKMHlOTG9WNjdMdTFnOGxwMjdmRXRsdVNJWXQ2WVE2
endQSTlZVWxCUFNGQW9ETWlfTjhvSGNxcXQ3VTFUdUpmMnlvUUtyVHdvbkxucG9kdVBxMktGV2RDMHVON2l
1VTQ4V3RyOTF1VDdGM0ZoT0w2c3c5bTR1RW42OGhNTFVPYWxSR3BkWF9feGZUeG5jc1lOSXZqSG5kLUFreW
JCNlJiRzJKRm1DLXlzZW1XVThpbk9aV09Qc21xcEJWOVloSGlXZFR2d05zeVlRNW05b1VvUy1qWUVWLXpxZ
0NtTFpsczdaT2RKRE9oOTR5UGZUZm1XSV9HYnh5S05ydS05aHhkZE5MTGRXaXBCRmVreTluX1BuanZtaFo2
bnRRaUtxRXJXMURPLTJmdDEtcW9tLUlxcWRGZGZpM3FzSXVlQ1BEM1puNDJ2Nm5URFk4ZXRZOVdmcUxra2J
Sdjl6eXRPMThrM285clNwSlh1dkZyU01lVmNKZTNPcmJHOTFpYmdfUF9XNU5pX1hZVndKOG0xT3FZLTJoQU
N6U3ltTXRXZWJkZkdRUmZCVi1heElMYi1NYktHdjBjcUI2VEhsOXVhQ0ptTXVvRkFkUFZmeV8yLTlWMjdpb
XRsLXJSd0lkbVNHWWgzUTN3aG44U196cnc3U2xpRW1VSDRUdzJpVTg5SWRWd1FndXd4dEcxVjhMb3VfMW1H
Vmt4XzhpTTFLSDhOeDNMOFBiT2JiZXBpZWJDYWVuamt1QnFWQ1dubl9pS0hrSXVQTHZTQ3B2am9PSUhNcTc
2WDZFUGtxU29oYl9QR2NhaVJNTTVHTkVxdHZVaVJ0RDNhRzZDVm1SSFk5VEQ1ZDFGQjJCZ01xYjRONGUza1
```

```
pLejFSbzY4dG5BX0pBSV9teFpRaU5fSWdHdw==","email":"cesarezuppa@cagroupsolutions.it"}}
}'
sshKey:                                                                    'ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIMXmXndMauIP8fxXpnyteyNitAP2LMOTKLhLazs/Qsp2
root@grpi-ocp-kv00'
```

E lanciata l'installazione tramite il seguente comando:

Unset

```
mkdir /root/ocp4-free-collaudo/installation_dir
cp install-config.yaml /root/ocp4-free-collaudo/installation_dir/.
cd /root/ocp4-free-collaudo

./openshift-install create cluster --dir=/root/ocp4-free-collaudo/installation_dir
--log-level debug
....

INFO Waiting up to 40m0s (until 4:01PM CET) for the cluster at
https://api.ocp-collaudo.cariprpccoll.it:6443 to initialize...
DEBUG Cluster is initialized
INFO Checking to see if there is a route at openshift-console/console...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/root/ocp4-free-collaudo/installation_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here:
https://console-openshift-console.apps.ocp-collaudo.cariprpccoll.it
INFO Login to the console with user: "kubeadmin", and password:
"spfjv-cvrQW-ia3Nj-MQoXS"
DEBUG Time elapsed per stage:
DEBUG pre-bootstrap: 37s
DEBUG bootstrap: 18s
DEBUG master: 17s
DEBUG Bootstrap Complete: 24m10s
DEBUG API: 4m31s
DEBUG Bootstrap Destroy: 2m18s
DEBUG Cluster Operators: 20m19s
INFO Time elapsed: 48m7s
```

Al termine è possibile verificare lo stato dei *clusterOperator* con il seguente comando:

Unset

```
oc get co

NAME                                       VERSION   AVAILABLE   PROGRESSING
DEGRADED   SINCE   MESSAGE
authentication                             4.14.2    True        False
False      2m36s
```

```
baremetal                                  4.14.2     True          False
False      29m
cloud-controller-manager                 4.14.2      True          False
False      36m
cloud-credential                           4.14.2     True          False
False      42m
cluster-autoscaler                         4.14.2     True          False
False      29m
config-operator                            4.14.2     True          False
False      30m
console                                    4.14.2     True          False
False      7m54s
control-plane-machine-set                4.14.2      True          False
False      29m
csi-snapshot-controller                  4.14.2      True          False
False      29m
dns                                          4.14.2     True          False
False      29m
etcd                                         4.14.2     True          False
False      28m
image-registry                             4.14.2     True          False
False      22m
ingress                                    4.14.2     True          False
False      10m
insights                                   4.14.2     True          False
False      23m
kube-apiserver                             4.14.2     True          False
False      22m
kube-controller-manager                  4.14.2      True          False
False      25m
kube-scheduler                             4.14.2     True          False
False      25m
kube-storage-version-migrator            4.14.2      True          False
False      30m
machine-api                                4.14.2     True          False
False      16m
machine-approver                           4.14.2     True          False
False      29m
machine-config                             4.14.2     True          False
False      28m
marketplace                                4.14.2     True          False
False      29m
monitoring                                 4.14.2     True          False
False      8m37s
network                                    4.14.2     True          False
False      29m
node-tuning                                4.14.2     True          False
False      29m
openshift-apiserver                        4.14.2     True          False
False      22m
openshift-controller-manager             4.14.2      True          False
False      25m
openshift-samples                          4.14.2     True          False
False      22m
operator-lifecycle-manager               4.14.2      True          False
False      29m
```

```
operator-lifecycle-manager-catalog                4.14.2       True            False
False      29m
operator-lifecycle-manager-packageserver      4.14.2        True               False
False      22m
service-ca                                            4.14.2     True           False
False      30m
storage                                               4.14.2     True           False
False      25m
```

E verificare lo stato dei nodi e delle machines:

Unset

```
oc get machines -A

NAME             PHASE    TYPE  REGION  ZONE  AGE
ocp-collaudo-jn6wt-master-0    Running             57m
ocp-collaudo-jn6wt-master-1    Running             57m
ocp-collaudo-jn6wt-master-2    Running             57m
ocp-collaudo-jn6wt-worker-0-8jkjr  Running             42m
ocp-collaudo-jn6wt-worker-0-gn5hn  Running             42m
ocp-collaudo-jn6wt-worker-0-q4mhl  Running             42m
ocp-collaudo-jn6wt-worker-0-q6q9j  Running             42m
ocp-collaudo-jn6wt-worker-0-vzt6w  Running             42m
ocp-collaudo-jn6wt-worker-0-x9rlj  Running             42m

oc get nodes

NAME                                 STATUS   ROLES                  AGE   VERSION
ocp-collaudo-jn6wt-master-0                  Ready       control-plane,master     50m
v1.27.6+f67aeb3
ocp-collaudo-jn6wt-master-1                  Ready       control-plane,master     52m
v1.27.6+f67aeb3
ocp-collaudo-jn6wt-master-2                  Ready       control-plane,master     51m
v1.27.6+f67aeb3
ocp-collaudo-jn6wt-worker-0-8jkjr     Ready      worker                           30m
v1.27.6+f67aeb3
ocp-collaudo-jn6wt-worker-0-gn5hn     Ready      worker                           32m
v1.27.6+f67aeb3
ocp-collaudo-jn6wt-worker-0-q4mhl     Ready      worker                           31m
v1.27.6+f67aeb3
ocp-collaudo-jn6wt-worker-0-q6q9j     Ready      worker                           29m
v1.27.6+f67aeb3
ocp-collaudo-jn6wt-worker-0-vzt6w     Ready      worker                           30m
v1.27.6+f67aeb3
ocp-collaudo-jn6wt-worker-0-x9rlj     Ready      worker                           28m
v1.27.6+f67aeb3
```

3. Day2 operations

Aggiunta nodi infrastrutturali

Per aggiungere i nodi infrastrutturali è stato preparato il seguenti template:

Unset

ocp-collaudo-jn6wt-infra-0.yaml

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
 name: ocp-collaudo-jn6wt-infra-0
 namespace: openshift-machine-api
spec:
 replicas: 6
 selector:
  matchLabels:
   machine.openshift.io/cluster-api-cluster: ocp-collaudo-jn6wt
   machine.openshift.io/cluster-api-machineset: ocp-collaudo-jn6wt-infra-0
 template:
  metadata:
   labels:
    machine.openshift.io/cluster-api-cluster: ocp-collaudo-jn6wt
    machine.openshift.io/cluster-api-machine-role: worker
    machine.openshift.io/cluster-api-machine-type: worker
    machine.openshift.io/cluster-api-machineset: ocp-collaudo-jn6wt-infra-0
  spec:
   lifecycleHooks: {}
   metadata:
    labels:
     node-role.kubernetes.io/infra: ""
    taints:
    - key: node-role.kubernetes.io/infra
     effect: NoSchedule
   providerSpec:
    value:
     apiVersion: machine.openshift.io/v1beta1
     credentialsSecret:
      name: vsphere-cloud-credentials
     diskGiB: 120
     kind: VSphereMachineProviderSpec
     memoryMiB: 32768
```

```
    metadata:
     creationTimestamp: null
    network:
     devices:
     - networkName: dvpg_619_DMZ_ocprhel_col
    numCPUs: 8
    numCoresPerSocket: 2
    snapshot: ""
    template: ocp-collaudo-jn6wt-rhcos-generated-region-generated-zone
    userDataSecret:
     name: worker-user-data
    workspace:
     datacenter: ACILIA
     datastore: /ACILIA/datastore/ESX-OCP-PREPROD-AC-0000
     folder: /ACILIA/vm/COLLAUDO/OCP
     resourcePool: /ACILIA/host/OCP_PREPROD//Resources
     server: ac-cags-vcsa001.cariprpc.it
```

E creati i nodi con il seguente comando:

```
Unset
oc apply -f ocp-collaudo-jn6wt-infra-0.yaml
```

Creazione MachineConfigPool

Per velocizzare eventuali operazioni che coinvolgono il riavvio controllato dei nodi openshift, è stato creato un machine set per i nodi infrastrutturali utilizzando il template seguente:

```
Unset
mcp-infra.yaml

apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfigPool
metadata:
  name: infra
spec:
  machineConfigSelector:
    matchExpressions:
    - key: machineconfiguration.openshift.io/role
      operator: In
      values:
      - worker
```

```
      - infra
  nodeSelector:
    matchExpressions:
    - key: node-role.kubernetes.io/infra
      operator: Exists
```

La creazione vera e propria avviene utilizzando il comando:

```
Unset
oc apply -f mcp-infra.yaml
```

Configurazione NTP

Per configurare i nodi affinchè utilizzassero l'NTP **10.213.10.28** e **10.213.10.138** , sono stati creati i seguenti template:

```
Unset
99-master-ntp.bu

variant: openshift
version: 4.10.0
metadata:
  name: 99-master-custom-ntp
  labels:
    machineconfiguration.openshift.io/role: master
storage:
  files:
   - path: /etc/chrony.conf
     mode: 0644
     overwrite: true
     contents:
       inline: |
         pool 0.rhel.pool.ntp.org iburst
         server 10.213.10.28
         server 10.213.10.138
         driftfile /var/lib/chrony/drift
         makestep 1.0 3
         rtcsync
         logdir /var/log/chrony

---
99-worker-ntp.bu
```

```
variant: openshift
version: 4.10.0
metadata:
  name: 99-worker-custom-ntp
  labels:
    machineconfiguration.openshift.io/role: worker
storage:
  files:
   - path: /etc/chrony.conf
     mode: 0644
     overwrite: true
     contents:
       inline: |
         pool 0.rhel.pool.ntp.org iburst
         server 10.213.10.28
         server 10.213.10.138
         driftfile /var/lib/chrony/drift
         makestep 1.0 3
         rtcsync
         logdir /var/log/chrony
```

E' stato scaricata l'utility "*butane*":

```
Unset
curl https://mirror.openshift.com/pub/openshift-v4/clients/butane/latest/butane
--output /usr/sbin/butane

chmod +x /usr/sbin/butane
```

E utilizzato per convertire i template con il seguente comando:

```
Unset
butane 99-worker-ntp.bu -o 99-worker-ntp.yaml
butane 99-master-ntp.bu -o 99-master-ntp.yaml
```

Le configurazioni sono state applicate sui nodi come segue:

```
Unset
oc appply -f 99-worker-ntp.yaml
oc appply -f 99-master-ntp.yaml
```

Spostamento monitoring su nodi infrastrutturali

Per spostare i pod del monitoring interno di OCP sui nodi infrastrutturali, è stato creato il seguente template che esplicita, componente per componente, il *nodeselector* infra e la *toleration* adatta, nonchè l'utilizzo di volumi persistenti per le componenti di *alertmanager* e *prometheus*:

Unset

cm-monitoring.yaml

```
apiVersion: v1
data:
 config.yaml: |
  enableUserWorkload: true
  alertmanagerMain:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
   volumeClaimTemplate:
    spec:
     storageClassName: thin-csi
     resources:
      requests:
       storage: 20Gi
  grafana:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
  k8sPrometheusAdapter:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
  kubeStateMetrics:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
  openshiftStateMetrics:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
```

```
  prometheusK8s:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
   volumeClaimTemplate:
    spec:
     storageClassName: thin-csi
     resources:
      requests:
       storage: 100Gi
  prometheusOperator:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
  thanosQuerier:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
  telemeterClient:
   nodeSelector:
    node-role.kubernetes.io/infra: ""
   tolerations:
   - effect: NoSchedule
    key: node-role.kubernetes.io/infra
kind: ConfigMap
metadata:
 name: cluster-monitoring-config
 namespace: openshift-monitoring
```

La seguente configurazione è stata messa applicata tramite il seguente comando:

```
Unset
oc apply -f cm-monitoring.yaml
```

Autenticazione

Crédit Agricole ha scelto di utlizzare l'IDAP, che normalmente utilizza sui propri sistemi, come metodo di autenticazione per l'ambiente OCP.

Per far questo è stata portata sul bastion la CA che firma il certificato utilizzato dal domain controller, *caldap.pem*, , che espone in ldaps, all'interno della directory /root/ocp4-free-collaudo/installation_dir/day2conf/

E' stata creata la *configmap* che contiene la CA, ad uso dell'oauth:

```
Unset
oc create configmap ca-config-map
--from-file=ca.crt=/root/ocp4-free-collaudo/installation_dir/day2conf/caldap.pem -n
openshift-config
```

E' stata creata la secret contenente la password di bind per l'utenza "CN=cp_dtr_ldap,OU=OU-UTENTI-SERVIZI,OU=OU-UTENTI,DC=cariprpccoll,DC=it" con il seguente comando:

```
Unset
oc create secret generic ldap-secret --from-literal=bindPassword=********** -n
openshift-config
```

A questo punto è stato configurato l'oggetto *oauth* del cluster utilizzando il seguente template:

```
Unset
oauth-cluster.yaml

apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  name: cluster
spec:
  identityProviders:
  - ldap:
      attributes:
        email:
        - mail
        id:
        - dn
        name:
        - cn
        preferredUsername:
        - sAMAccountName
      bindDN:
CN=cp_dtr_ldap,OU=OU-UTENTI-SERVIZI,OU=OU-UTENTI,DC=cariprpccoll,DC=it
      bindPassword:
```

```
        name: ldap-secret
      ca:
        name: ca-config-map
      insecure: false
      url:
ldaps://msad0coll.cariprpccoll.it/DC=cariprpccoll,DC=it?sAMAccountName?sub?(&(objec
tClass=user)(|(memberof=CN=Users,DC=cariprpccoll,DC=it)(memberof=CN=GU_UCP_ADMIN,CN
=Users,DC=cariprpccoll,DC=it)(memberof=CN=GU_UCP_USER,CN=Users,DC=cariprpccoll,DC=i
t)(memberof=CN=GU_DTR_USER,CN=Users,DC=cariprpccoll,DC=it)(memberof=CN=GU_OCP_ADMIN
,CN=Users,DC=cariprpccoll,DC=it)(memberof=CN=GU_OCP_USER,CN=Users,DC=cariprpccoll,D
C=it)))
    mappingMethod: claim
    name: ldap
    type: LDAP
```

IMPORTANTE:

Nel caso in cui si vogliano aggiungere/escludere alcuni gruppi LDAP abilitati alla login di OCP, andrà eseguito il comando “oc edit oauth cluster” e modificato il campo `url` aggiungendo/escludendo il gruppo dal filtro.

Sync utenti e gruppi LDAP

Per avere sempre un sync continuo tra gli utenti e i gruppi definiti sull’LDAP e quelli definiti sull’ambiente OCP, si utilizza *l’operator* chiamato “Group Sync Operator”.

Tale operator si installa direttamente dalla dashboard OCP, dopo aver eseguito una ricerca all’interno della sezione “Operator Hub”.

Una volta installato occorre definire:

- La secret dell’utenza di bind:

```
Unset
oc create secret generic ldap-creds-group-sync
--from-literal=username="CN=cp_dtr_ldap,OU=OU-UTENTI-SERVIZI,OU=OU-UTENTI,DC=caripr
pccoll,DC=it" --from-literal=password='XXXXXXXXX' -n group-sync-operator
```

- Inserire la CA a firma del certificato utilizzato dall’LDAP, all’interno delle secret utilizzabili dal group sync operator:

```
Unset
oc create secret generic ldap-ca-bundle-group-sync
--from-file=ca.crt=/root/ocp4-free-collaudo/installation_dir/day2conf/caldap.pem -n
group-sync-operator
```

A questo punto si può utilizzare il seguente template:

```
Unset
groupsync-operator.yaml

apiVersion: redhatcop.redhat.io/v1alpha1
kind: GroupSync
metadata:
 name: ldap-groupsync
spec:
 providers:
 - ldap:
   credentialsSecret:
    name: ldap-creds-group-sync
    namespace: group-sync-operator
   caSecret:
    name: ldap-ca-bundle-group-sync
    namespace: group-sync-operator
   insecure: false
   activeDirectory:
    userNameAttributes:
    - sAMAccountName
    groupMembershipAttributes:
    - memberOf
    usersQuery:
     baseDN: DC=cariprpccoll,DC=it
     derefAliases: never
     scope: sub
     pageSize: 0
     filter:
(&(objectClass=person)(|(memberof=CN=Users,DC=cariprpccoll,DC=it)(memberof=CN=GU_UC
P_ADMIN,CN=Users,DC=cariprpccoll,DC=it)(memberof=CN=GU_UCP_USER,CN=Users,DC=cariprp
ccoll,DC=it)(memberof=CN=GU_DTR_USER,CN=Users,DC=cariprpccoll,DC=it)(memberof=CN=GU
_OCP_ADMIN,CN=Users,DC=cariprpccoll,DC=it)(memberof=CN=GU_OCP_USER,CN=Users,DC=cari
prpccoll,DC=it)))
   url: ldaps://msad0coll.cariprpccoll.it
   whitelist:
    - CN=Users,DC=cariprpccoll,DC=it
    - CN=GU_UCP_ADMIN,CN=Users,DC=cariprpccoll,DC=it
    - CN=GU_UCP_USER,CN=Users,DC=cariprpccoll,DC=it
    - CN=GU_DTR_USER,CN=Users,DC=cariprpccoll,DC=it
    - CN=GU_OCP_ADMIN,CN=Users,DC=cariprpccoll,DC=it
    - CN=GU_OCP_USER,CN=Users,DC=cariprpccoll,DC=it
  name: ldap-group-sync
```

Ed avviare la creazione tramite il comando:

```
Unset
oc apply -f groupsync-operator.yaml
```

IMPORTANTE:

Nel caso in cui si vogliano aggiungere/escludere alcuni gruppi LDAP, andrà eseguito il comando “oc edit GroupSync ldap-groupsync -n group-sync-operator ” e modificare sia il campo `whitelist` che il campo `filter` aggiungendo/escludendo il gruppo.

Per un corretto funzionamento dell’autenticazione e del sync dei gruppi, è necessario tenere allineate le configurazioni di Oauth e GroupSync a livello di gruppi definiti.

Assegnazione ruoli

Per assegnare un cluster role ad un gruppo eseguire il seguente comando:

```
Unset
oc adm policy add-cluster-role-to-group cluster-admin
CN=GU_OCP_ADMIN,CN=Users,DC=cariprpccoll,DC=ita
```

Segue una tabella riepilogativa dei cluster role definiti su OCP:

Default cluster role	Description
`admin`	A project manager. If used in a local binding, an `admin` has rights to view any resource in the project and modify any resource in the project except for quota.
`basic-user`	A user that can get basic information about projects and users.
`cluster-admin`	A super-user that can perform any action in any project. When bound to a user with a local binding, they have full control over quota and every action on every resource in the project.
`cluster-status`	A user that can get basic cluster status information.
`cluster-reader`	A user that can get or view most of the objects but cannot modify them.
`edit`	A user that can modify most objects in a project but does not have the power to view or modify roles or bindings.
`self-provisioner`	A user that can create their own projects.
`view`	A user who cannot make any modifications, but can see most objects in a project. They cannot view or modify roles or bindings.

Per applicarli eseguire i seguenti comandi:

```
Unset
oc get group

NAME ....

oc adm add-cluster-role-to-group CLUSTER-ROLE GROUP-NAME
```

Backup etcd

E' stato creato uno script che esegue, su base giornaliera, il backup dell'etcd e tiene in linea sul nodo bastion gli ultimi 5 giorni di backup.

Lo script è il seguente:

```
Unset
#!/bin/bash
/bin/echo [$(date +"%F %T")] Starting OCP Backup... &>> /var/log/ocp-backup.log
/bin/ssh -i /root/.ssh/ocp4key core@10.215.86.26 '/bin/sudo
/usr/local/bin/cluster-backup.sh /home/core/backup && /bin/sudo /bin/find
/home/core/backup -mtime +5 -delete && /bin/sudo /bin/chown -vR core:core
/home/core/backup'
/bin/rsync -av --delete -e "/bin/ssh -i /root/.ssh/ocp4key"
core@10.215.86.26:/home/core/backup /root/backup-etcd &>> /var/log/ocp-backup.log
/bin/echo [$(date +"%F %T")] Terminated OCP Backup. &>> /var/log/ocp-backup.log
```

Tale script è stato aggiunto al cron dell'utente root tramite il seguente comando:

```
Unset
crontalb -e

...

46 1 * * * /usr/local/sbin/backup-etcd-free-collaudo.sh
```

Modifica default ingress certificate

Per modificare il certificato di default dell'ingress, per la wildcard *apps, eseguire i seguenti passi:

1. Creare una *configmap* che contenga il *rootCa* certificate utilizzato per firmare il certificato *appps…
2. Aggiungere all'oggeto *proxy/cluster* il riferimento a tale CA
3. Creare una secret che contenga i riferimenti al certificato della wildcard e alla sua chiave privata
4. Aggiungere all'oggetto *ingresscontroller* il riferimento a tale secret.

Per i comandi precisi si rimanda alla documentazione ufficiale:
https://docs.openshift.com/container-platform/4.14/security/certificates/replacing-default-ingress-certificate.html

ATTENZIONE:

Fare attenzione ai prerequisiti prima di eseguire la seguente attività:

- E' necessario avere un certificato wildcard e la corrispondente chiave privata per tutti i sottodomini .apps ognuno in formato PEM.
- La chiave privata non deve essere criptata, se lo è va decriptata precedentemente.
- Il certificato deve contenere *.apps.<clustername>.<domain> nell'estensione subjectAltName.
- Nel caso di catena di certificati (chain) l'ordine dei certificati nel file è importante: il primo dev'essere il certificato della wildcard, poi qualsiasi certificato intermediate, l'ultimo deve essere la root CA.
- La root CA deve essere in formato PEM.

Modifica certificato API

Per modificare il certificato delle api di OCP, eseguire i seguenti passi:

1. Creare una secret che contenga il certificato dell'fqdn delle (api…) e la chiave privata
2. Aggiungere all'oggeto apiserver il riferimento a tale secret

Per i comandi precisi si rimanda alla documentazione ufficiale:
https://docs.openshift.com/container-platform/4.14/security/certificates/api-server.html

ATTENZIONE:

Fare attenzione ai prerequisiti prima di eseguire la seguente attività:

- E' necessario avere un certificato per l'FQDN delle api e la corrispondente chiave privata ognuno in formato PEM.
- La chiave privata non deve essere criptata, se lo è va decriptata precedentemente.
- Il certificato deve contenere api.<clustername>.<domain> nell'estensione subjectAltName.
- Nel caso di catena di certificati (chain) l'ordine dei certificati nel file è importante: il primo dev'essere il certificato dell'FQDN delle api, poi qualsiasi certificato intermediate, l'ultimo deve essere la root CA.