Red Hat Consulting

Crédit Agricole Group Solution

Advanced Cluster Management for Kubernetes v2.10.2

Documento di installazione

Ambiente di Produzione

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.

Red Hat Consulting

Review History

Version	Date	Contributor	Role	Description
1.0	16/05/2024	Giovanni Filice Domenico Pastore Matteo Santucci Andrea Tozzoli	Cloud Consultant Cloud Consultant Cloud Architect Project Manager	First draft
1.1	17/05/2024	Giovanni Filice Andrea Tozzoli	Cloud Consultant Project Manager	- Added cap. “2.3.7 Aumento replica IngressController a 3” - Correction of some typos

Red Hat Consulting

Table of Contents

1. Introduzione **5**
1.1. Purpose 5
1.2. Termini e acronimi 5
2. Cluster HUB di management **7**
2.1. Installazione cluster base Acilia 7
2.2. Installazione ACM Acilia 14
2.3. Configurazioni aggiuntive 15
2.3.1. Servizio Chronyd 16
2.3.2. Definizione nodi infrastrutturali 22
2.3.3. Autenticazione tramite LDAP 26
2.3.4. Sync gruppi di utenti tra alberatura LDAP e OCP 30
2.3.5. Installazione ODF 37
2.3.6. Configurazione ODF 37
2.4. Installazione cluster base Rozzano 38
2.5. Installazione ACM Rozzano 45
2.6. Configurazioni aggiuntive 46
2.6.1. Servizio Chronyd 47
2.6.2. Definizione nodi infrastrutturali 52
2.6.3. Autenticazione tramite LDAP 56
2.6.4. Sync gruppi di utenti tra alberatura LDAP e OCP 60
2.6.5. Installazione ODF 66
2.6.6. Configurazione ODF 66
3. Configurazione Active Passive **67**
3.1. Policy 67
3.2. Configurazione CA-Bundle 68
3.3. Configurazione Bucket S3 69
3.4. Configurazione Acilia come active cluster 70
3.5. Configurazione Rozzano come passive cluster 77
4. Procedura di Failover **81**
5. Procedura di Failback **83**

1. Introduzione

Red Hat è stata ingaggiata da Crédit Agricole per installare l'ambiente di Produzione di ACM..

Tale ambiente è costituito dai seguenti cluster:

- Un cluster di management che ospita la componente ACM, installato nel DC del sito di Acilia e identificato con il nome **ocp-acmac**.
- Un cluster di management che ospita la componente ACM, installato nel DC del sito di Rozzano e identificato con il nome **ocp-acmrz**.

Questo documento tratta l'installazione dei due cluster, le configurazioni dei vari servizi a contorno e la configurazione in active-passive dei 2 cluster ACM..

La modalità di installazione scelta per questi ambienti è la "IPI", per uniformità con gli ambienti inferiori la versione di OpenShift Container Platform scelta è la **4.14.2**, la piattaforma sottostante è il vSphere e la versione di ACM configurata è la **2.10.2**.

1.1. Purpose

Questo documento descrive la procedura di installazione dei cluster, in modalità IPI, e le configurazioni apportate ai vari servizi compresi.

1.2. Termini e acronimi

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

Acronym	Description
ACM	Red Hat Advanced Cluster Management for Kubernetes
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	Red Hat OpenShift Container Platform
ODF	Red Hat Openshift Data Foundation
RH	Red Hat, Inc
RHEL	Red Hat Enterprise Linux
SAN	Storage Area Network
SSL	Secure Sockets Layer
VIP	Virtual IP address
VLAN	Virtual LAN is a networking virtualisation technology
Workload	Synonym for "Guest" or container running on OCP

2. Cluster HUB di management

2.1. Installazione cluster base Acilia

La piattaforma che ospita il cluster OCP di management è un vSphere in versione 7.0.3 , compatibile con la versione di OCP scelta, la 4.14.2.

Tale cluster è composto dalle seguenti macchine virtuali:

Ruolo	vCPU	RAM	OS disk	Count
Master	16	32	120 GB	3
Infra	16	32	120 GB	3

Come attività preliminari all'installazione sono stati eseguiti i seguenti task:

- Impostazione del proxy sul bastion **GRPI-OCP-PV00**
- Creazione della directory di lavoro **/root/ocp-acmac**
- Creazione chiave ssh per l'accesso ai nodi OCP **/root/.ssh/ocp-produzione**
- Download client e installer in versione **4.14.2**
- Copia in locale dei certificati del vCenter
- Verifica/Download della Pull Secrets dal Portale RedHat

Creare Directory di lavoro

```
Python
[root@GRPI-OCP-PV00 ~]# mkdir /root/ocp-acmrz
```

```
Python
[root@GRPI-OCP-PV00 ~]#  cd ocp-acmrz
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# mkdir install_dir
```

Impostare i proxy

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# export
https_proxy=http://vip-navproxy-server.cariprpc.it:8080
[root@GRPI-OCP-PV00 ocp-acmrz]# export
http_proxy=http://vip-navproxy-server.cariprpc.it:8080
[root@GRPI-OCP-PV00 ocp-acmrz]# export
no_proxy=localhost,127.0.0.1,localaddress,.localdomain.com,.cariprpc.it,10.68.0.0/1
4,172.27.0.0/16,10.215.87.0/24
```

Scaricare Client e Installer di Openshift

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# wget
https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.14.2/openshift-i
nstall-linux.tar.gz

[root@GRPI-OCP-PV00 ocp-acmrz]# wget
https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.14.2/openshift-c
lient-linux-4.14.2.tar.gz
```

Scompattare il Client e Installer

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# tar zxvf openshift-install-linux.tar.gz

[root@GRPI-OCP-PV00 ocp-acmrz]# chmod +x /root/ocp-acmrz/openshift-install
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# tar zxvf openshift-install-linux.tar.gz
[root@GRPI-OCP-PV00 ocp-acmrz]# chmod +x /usr/bin/oc
[root@GRPI-OCP-PV00 ocp-acmrz]# tar zxvf openshift-client-linux-4.14.2.tar.gz -C
/usr/local/sbin
```

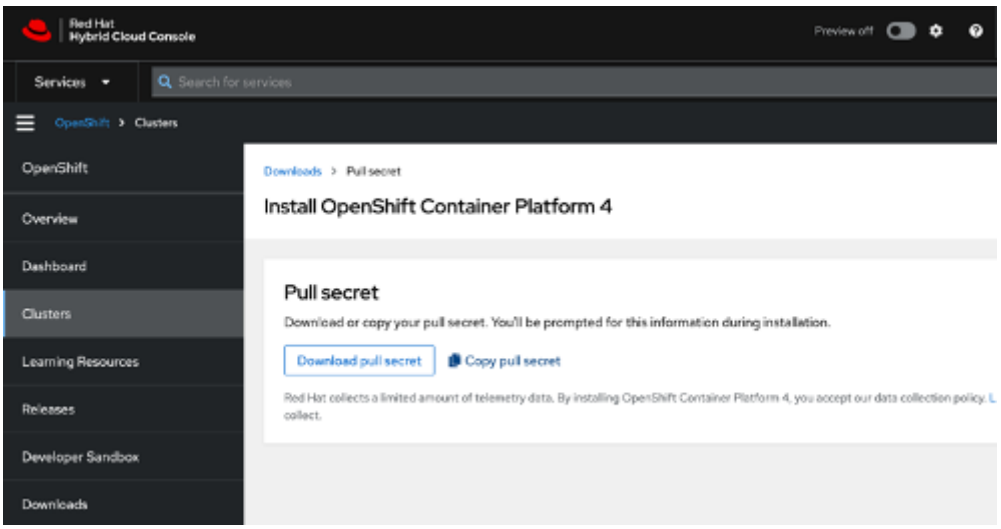
Scaricare e 'Trustare' i certificati di Vmware

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz ~]# cd /root

[root@GRPI-OCP-PV00 ocp-acmrz ~]# wget
https://ac-cags-vcsa001.cariprpc.it/certs/download.zip

[root@GRPI-OCP-PV00 ocp-acmrz]# cp /root/certs/lin/*
/etc/pki/ca-trust/source/anchors

[root@GRPI-OCP-PV00 ocp-acmrz]# update-ca-trust
```

Generare Chiave SSH

```
Python
[root@GRPI-OCP-PV00GRPI-OCP-PV00 ocp-acm]# ssh-keygen -t ed25519 -N '' -f
/root/.ssh/ocp-produzione

Generating public/private ed25519 key pair.
Your identification has been saved in /root/.ssh/ocp-produzione
Your public key has been saved in /root/.ssh/ocp-produzione.pub
The key fingerprint is:
SHA256:LgYRq643Ll3Hzeti/Z7EkdxVEgq2NBX//1kSuvh+aSc root@
The key's randomart image is:
+--[ED25519 256]--+
|      .      =.o.o..|
|      o   o + o o |
|      o      . . o  |
|   . .   . o . . |
|  . .. oS + . . .|
| .  ..o.o. . . ..|
| ... .o...o . ..o|
|..+  .o.o. o .E.=|
|.+.. . o.o=o+o +.|
+----[SHA256]-----+
[root@GRPI-OCP-PV00 ocp-acm]# eval "$(ssh-agent -s)"
Agent pid 54467

[root@GRPI-OCP-PV00 ocp-acm]# ssh-add /root/.ssh/ocp-produzione
Identity added: /root/.ssh/ocp-produzione (root@)
```

Per poter installare correttamente OCP il file di configurazione chiamato install-config necessita delle credenziali di accesso al registry di Red Hat, tale credenziali è chiamata “pull Secrets” ed è scaricabile al seguente link



A questo punto è stato preparato l’*install-config.yaml* che riportiamo:

```python
apiVersion: v1
baseDomain: cariprpc.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,.cariprpc.it,.cariprpcpar.it,.car
iprpccoll.it,172.30.0.0/16,10.19.84.0/22,10.19.87.5,10.19.87.4
compute:
- name: worker
  hyperthreading: Enabled
  platform:
    vsphere:
      cpus: 16
      coresPerSocket: 2
      memoryMB: 32768
      osDisk:
        diskSizeGB: 120
  replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
```

```
    vsphere:
      cpus: 16
      coresPerSocket: 2
      memoryMB: 32768
      osDisk:
        diskSizeGB: 120
  replicas: 3
metadata:
  name: ocp-acmac
networking:
  machineNetwork:
  - cidr: 10.19.84.0/22
  networkType: OVNKubernetes
  serviceNetwork:
  - 172.30.0.0/16
platform:
  vsphere:
    apiVIP: 10.19.87.5
    cluster: "OCP_PROD"
    datacenter: "ACILIA"
    defaultDatastore: ESX-OCP-PROD-AC-0000
    ingressVIP: 10.19.87.4
    network: "dvpg_3_MGMT_AC"
    password: 'N65sCPnrD$AasJANJThc'
    username: cariprpc\cp_12_vcenter_OCP
    vCenter: ac-cags-vcsa001.cariprpc.it
    folder: "/ACILIA/vm/PRODUZIONE/OCP"
publish: External
pullSecret:
'{"auths":{"cloud.openshift.com":{"auth":"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc
3NfZTdjYzcwZDhkNjZlNGE1MTgxYWZlOTAzYTNiMTI2NDk6TjZHSFpKUTI4RFExNEgzT05PVlQ2UTZYUU4z
MU40UU9TVzRYTFNEVjdUOElBVDA3V01VTFpVSFBFQzRYTkFSVg==","email":"daniele.bagiotti@cre
dit-agricole.it"},"quay.io":{"auth":"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfZT
djYzcwZDhkNjZlNGE1MTgxYWZlOTAzYTNiMTI2NDk6TjZHSFpKUTI4RFExNEgzT05PVlQ2UTZYUU4zMU40U
U9TVzRYTFNEVjdUOElBVDA3V01VTFpVSFBFQzRYTkFSVg==","email":"daniele.bagiotti@credit-a
gricole.it"},"registry.connect.redhat.com":{"auth":"fHVoYy1wb29sLTgxNGFiODI0LTgwMzc
tNGJjMy1iMTA2LWUwZDQyMWY5NDU2ZjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSTJNekZpWW
1Ka09XTXlOVGswWTJaak9UWm1abUkwWVRRMU4yTmhORE00TVNKOS5jdVU0TjA4R3Q5VUh6Snk0d3VXOV9ae
FdTc1FiN0dUYThWRm0yTTdqWFBmSXd3TXZ0YnZGTHVwYno5bTZZX3VXTE91UF9SZk9xSVpldGk1X3oxUnBN
OFVnRDMzUUJ2V2QxbGFxMkJEY0ZUTHlyQ2Z5anVvaGdvZFppZE5SRFgwQXRURFdDLTlOMmRlMTZMUWhaeUt
tYnBaZlFDVVo5TGtDLWdQSk1xVmNTOFY3RzhXM2x0QWkyTm55TGp1dzBDMk5MUjl3M3R4M29tRnhES1Jabm
1kYmFQemdzQ2o4bnNCZXhDYmtYRFVCVWo1YUFjX1Fha3FxNVZZYmtxYTQtNHRZYTljamc3RXEtUy11dW5pW
lo3cldxLUhOdmlYRUpWVVY0M1NkdnlNZ2I0bTQwanFaYVpqOVpLQ1pycktVUdnd0tTS1IteUNwcHhWZWZS
cnB0QmZ4WndrM0ZPNGg1azZuOUdOSE1RazdmRGVnTEJ5WVdmcVdQNnZHSENudW1LRVFBLWczcXBPZENrWlp
tV2VRQlpHaExHNkhpS0luRHVIOUxKb2hhSm5BcTBqVVhsVHdGYWZ4OUtoVXhUemM1cDhBUmQwZGdEd2tUZj
JaVWZKM2czVW1QUjNHaTRmY2xTaHc5RW9UbXlKXzYzdHpSV0pJd053X0VSOE9QSHVnNFNwM2pfZ2lPaXNCS
mI2TU5ybkJ3WmFXWElDNjdCbnR2REJVTXgyZkdYbDJ2cDZEamdpWjZxU2Z5blRwWXFpaFBGSWplbldQd0Vq
UFl5b01nQTF2NVJoanppT1ZCemZXRTVKY1pOdC1VaU5GckdaMmV0bXdQUkpNaThJODJ2YWx3SjlxYkNZT21
fQ2xsRWI4TEF0aWJDTGtyYUVPRmY2cmpyMGRSUXZ1V1RIb3BJTQ==","email":"daniele.bagiotti@cr
edit-agricole.it"},"registry.redhat.io":{"auth":"fHVoYy1wb29sLTgxNGFiODI0LTgwMzctNG
JjMy1iMTA2LWUwZDQyMWY5NDU2ZjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSTJNekZpWW1Ka
09XTXlOVGswWTJaak9UWm1abUkwWVRRMU4yTmhORE00TVNKOS5jdVU0TjA4R3Q5VUh6Snk0d3VXOV9aeFdT
c1FiN0dUYThWRm0yTTdqWFBmSXd3TXZ0YnZGTHVwYno5bTZZX3VXTE91UF9SZk9xSVpldGk1X3oxUnBNOFV
nRDMzUUJ2V2QxbGFxMkJEY0ZUTHlyQ2Z5anVvaGdvZFppZE5SRFgwQXRURFdDLTlOMmRlMTZMUWhaeUttYn
BaZlFDVVo5TGtDLWdQSk1xVmNTOFY3RzhXM2x0QWkyTm55TGp1dzBDMk5MUjl3M3R4M29tRnhES1Jabm1kY
mFQemdzQ2o4bnNCZXhDYmtYRFVCVWo1YUFjX1Fha3FxNVZZYmtxYTQtNHRZYTljamc3RXEtUy11dW5pWlo3
cldxLUhOdmlYRUpWVVY0M1NkdnlNZ2I0bTQwanFaYVpqOVpLQ1pycktVUdnd0tTS1IteUNwcHhWZWZScnB
```

```
0QmZ4WndrM0ZPNGg1azZuOUdOSE1RazdmRGVnTEJ5WVdmcVdQNnZHSENudW1LRVFBLWczcXBPZENrWlptV2
VRQlpHaExHNkhpS0luRHVIOUxKb2hhSm5BcTBqVVhsVHdGYWZ4OUtoVXhUemM1cDhBUmQwZGdEd2tUZjJaV
WZKM2czVW1QUjNHaTRmY2xTaHc5RW9UbXlKXzYzdHpSV0pJd053X0VSOE9QSHVnNFNwM2pfZ2lPaXNCSmI2
TU5ybkJ3WmFXWElDNjdCbnR2REJVTXgyZkdYbDJ2cDZEamdpWjZxU2Z5blRwWXFpaFBGSWplbldQd0VqUFl
5b01nQTF2NVJoanppT1ZCemZXRTVKY1pOdC1VaU5GckdaMmV0bXdQUkpNaThJODJ2YWx3SjlxYkNZT21fQ2
xsRWI4TEF0aWJDTGtyYUVPRmY2cmpyMGRSUXZ1VlRIb3BJTQ==","email":"daniele.bagiotti@credi
t-agricole.it"}}}'
sshKey: 'ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIAG9AnCEFAut5NP+i5mHwh08c/0qTwLWRTZXIZwhhW+Z
root@GRPI-OCP-PV00'
```

L'*install-config.yaml* è stato posizionato sia all'interno della directory **/root/ocp-acmac** che **/root/ocp-acm/install_dir.** L'installazione è stata lanciata con il comando seguente:

Python

```
[root@GRPI-OCP-PV00 ocp-acmac]# cp install-config.yaml ./install_dir/.

[root@GRPI-OCP-PV00 ocp-acmac]# ./openshift-install create cluster --dir
/root/ocp-acmac/install_dir --log-level  debug
....
INFO Checking to see if there is a route at openshift-console/console...
DEBUG Route found in openshift-console namespace: console
DEBUG OpenShift console route is admitted
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/root/ocp-acm/installation_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here:
https://console-openshift-console.apps.ocp-acmac.cariprpc.it
INFO Login to the console with user: "kubeadmin", and password: "XXXXXXXXXXX"
DEBUG Time elapsed per stage:
DEBUG      pre-bootstrap: 38s
DEBUG          bootstrap: 12s
DEBUG             master: 17s
DEBUG Bootstrap Complete: 24m12s
DEBUG                API: 5m30s
DEBUG  Bootstrap Destroy: 1m14s
DEBUG  Cluster Operators: 18m22s
INFO Time elapsed: 45m2s
[root@grpi-ocp-hv00 ocp-acm]#
```

Al termine dell'installazione è possibile settare la variabile di ambiente KUBECONFIG per autenticarsi sul cluster appena creato:

```
Python
[root@GRPI-OCP-PV00 ocp-acmac]# export
KUBECONFIG=/root/ocp-acm/install_dir/auth/kubeconfig
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmac]# oc whoami
```

ATTENZIONE: Dato che sarà utilizzato lo stesso Bastion anche per la successiva installazione del cluster di Rozzano abbiamo preferito creare un alias contenente il suddetto comando di export

Modificare il seguente file:

```
Python
[root@GRPI-OCP-PV00 ocp-acmac]# vi ~/.bashrc
```

Aggiungere le seguenti entry:

```
Python
alias ocp-acmac=export KUBECONFIG=/root/ocp-acmac/install_dir/auth/kubeconfig
alias ocp-acmrz=export KUBECONFIG=/root/ocp-acmrz/install_dir/auth/kubeconfig
```

Ricaricare la sessione Bash

```
Python
[root@GRPI-OCP-PV00 ocp-acmac]# source ~/.bashrc
```

Collegarsi al cluster

```
Python
[root@GRPI-OCP-PV00]# ocp-acmac
```

Al termine della fase di installazione avremo un cluster composto da 3 nodi master e da 3 nodi worker.
In una fase successiva andremo a sostituire i nodi worker con dei nodi infrastrutturali.

La corretta installazione del cluster può essere verificata con il seguente comando

```
Python
[root@GRPI-OCP-PV00 ocp-acmac]# oc get clusteroperators
```

L'output conterrà lo stato del cluster e la colonna "AVAILABLE" dovrà essere "True"

2.2. Installazione ACM Acilia

Al termine dell'installazione minimale è stato installato l'operator "*Advanced Cluster Management for Kubernetes*" utilizzando la dashboard.

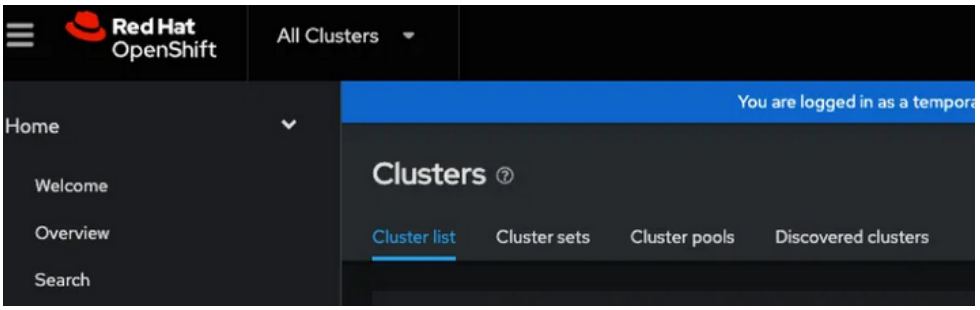
I passi eseguiti sono i seguenti:

- Collegandosi alla console di OCP del cluster OCP-ACMAC, all'interno del tab "Operator Hub", digitare "Advanced Cluster Management for Kubernetes" e cliccare su "install".
- Nella pagina della subscription che si apre, lasciare il default "open-cluster-management" come namespace di default per l'installazione.
- Modalità di aggiornamento: selezionare "Automatic".

Una volta installato l'operator, selezionarlo e modificare la strategia di aggiornamento della subscription, da "Automatic" a "Manual".

Cliccando all'interno della sezione "MultiClusterHub" creare l'oggetto "MultiClusterHub" lasciando i valori di default.

Terminata l'installazione, è possibile collegarsi alla console di ACM utilizzando il tab "All Clusters" che si trova in alto a sinistra della console di OCP.



2.3. Configurazioni aggiuntive

Le cosiddette “operazioni di Day2”, sono state eseguite sul cluster HUB ACM utilizzando le policy di ACM.

Per semplificare la definizione e la gestione delle policy, è stato utilizzato il PolicyGenerator.

Seguono i comandi eseguiti per configurare il plugin del PolicyGenerator sul nodo bastion:

```
Python
[root@GRPI-OCP-PV00 ocp-acmac]# mkdir -p ${HOME}/.config/kustomize/plugin/policy.open-cluster-management.io/v1/policygenerator
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmac ~]# wget https://github.com/open-cluster-management-io/policy-generator-plugin/releases/download/v1.13.0/linux-amd64-PolicyGenerator
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmac]# chmod +x linux-amd64-PolicyGenerator
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmac]# mv linux-amd64-PolicyGenerator
${HOME}/.config/kustomize/plugin/policy.open-cluster-management.io/v1/policygenerator/PolicyGenerator
```

All'interno della directory "**/root/cluster-acm-policy-generator/acm-hub/ocp-acmac**" sono state create le policy specifiche per il cluster ocp-acmac, nei prossimi paragrafi verranno descritte puntualmente.

Inoltre all'interno della directory "**/root/cluster-acm-policy-generator/acm-hub/all-cluster**" sono state create le policy valide sia per il cluster **ocp-acmac** che per il cluster **ocp-acmrz** quali, ad esempio, NTP ,Ldap sync, Oauth e ODF Operators.

2.3.1. Servizio Chronyd

All'interno della directory "**all-cluster/ntp"** sono stati definiti i template necessari a configurare il servizio *chronyd* sulle macchine virtuali di OCP.

Posizionarsi nella cartella contenente la policy "**/root/cluster-acm-policy-generator/acm-hub/all-cluster/ntp"**

```
Python
[root@GRPI-OCP-PV00 ntp]# cd
/root/cluster-acm-policy-generator/acm-hub/all-cluster/ntp
```

```
Python
[root@GRPI-OCP-PV00 ntp]# ll

-rw-r--r--. kustomization.yaml
-rw-r--r--. ntp-conf.yaml
-rw-r--r--. ntp-master-conf.yaml
-rw-r--r--. ntp-worker-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 ntp]# cat kustomization.yaml

generators:
  - ntp-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 ntp]# cat ntp-conf.yaml

apiVersion: policy.open-cluster-management.io/v1
kind: PolicyGenerator
metadata:
  name: generator-ntp-conf-ocp
placementBindingDefaults:
  name: placement-binding-ntp-conf-ocp
policyDefaults:
  namespace: acm-hub-policy
  placement:
    clusterSelectors:
      name: local-cluster
  complianceType: musthave
  remediationAction: enforce
  severity: high
policies:
  - name: policy-ntp-master-ocp
    manifests:
      - path: ntp-master-conf.yaml
  - name: policy-ntp-worker-ocp
    manifests:
      - path: ntp-worker-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 ntp]# cat ntp-master-conf.yaml

# Generated by Butane; do not edit
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: master
  name: 99-master-custom-ntp
spec:
  config:
    ignition:
      version: 3.4.0
    storage:
```

```
      files:
        - contents:
            compression: gzip
            source:
data:;base64,H4sIAAAAAAAAC/3zLMQ7CMAwAwN2v8AsSShlgRGJl4gWpmxaLUEeOQervkVDFhDyfrooUvN
7Oly5QUq5aKbAhDy9tBj/du9q7enD16Orpn47Kk01cMsZ30lh4iHRXWdb4FXimR26WK3Zhhz2oUVsXgiLzy
LodmbcDnwAAAP//yNra1AIBAAA=
          mode: 420
          overwrite: true
          path: /etc/chrony.conf
```

Python

```
[root@GRPI-OCP-PV00 ntp]# cat ntp-worker-conf.yaml

# Generated by Butane; do not edit
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-custom-ntp
spec:
  config:
    ignition:
      version: 3.4.0
    storage:
      files:
        - contents:
            compression: gzip
            source:
data:;base64,H4sIAAAAAAAAC/3zLMQ7CMAwAwN2v8AsSShlgRGJl4gWpmxaLUEeOQervkVDFhDyfrooUvN
7Oly5QUq5aKbAhDy9tBj/du9q7enD16Orpn47Kk01cMsZ30lh4iHRXWdb4FXimR26WK3Zhhz2oUVsXgiLzy
LodmbcDnwAAAP//yNra1AIBAAA=
          mode: 420
          overwrite: true
          path: /etc/chrony.conf
```

N.B: Butane (precedentemente Fedora CoreOS Config Transpiler) è uno strumento che 'legge' un file con sintassi Butane Config e produce una Ignition Config e verrà utilizzato per generare i MachineConfig di Openshift.

I file **ntp-master-conf.yaml** e **ntp-worker-conf.yaml** sono stati generati con l'utility "butane" come segue:

Scaricare il sorgente di Butane

```
Python
[root@GRPI-OCP-PV00 ntp]# curl
https://mirror.openshift.com/pub/openshift-v4/clients/butane/latest/butane --output
butane
```

Concedere i permessi di esecuzione

```
Python
[root@GRPI-OCP-PV00 ntp]# chmod +x butane
```

Spostarlo nella cartella /usr/local/sbin per renderlo utilizzabile sul bastion

```
Python
[root@GRPI-OCP-PV00 ntp]# mv butane /usr/local/sbin
```

Creare il file ntp-master.bu

```
Python
[root@GRPI-OCP-PV00 ntp]# cat ntp-master.bu

variant: openshift
version: 4.14.0
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 MSAD1.cariprpc.it iburst
```

```
        pool MSAD2.cariprpc.it iburst
        pool MSAD3.cariprpc.it iburst
        pool MSAD4.cariprpc.it iburst
        pool MSAD8.cariprpc.it iburst
        pool MSAD9.cariprpc.it iburst
        driftfile /var/lib/chrony/drift
        makestep 1.0 3
        rtcsync
        logdir /var/log/chrony
```

Creare il file ntp-worker.bu

```
Python
[root@GRPI-OCP-PV00 ntp]# cat ntp-worker.bu

variant: openshift
version: 4.14.0
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 MSAD1.cariprpc.it iburst
          pool MSAD2.cariprpc.it iburst
          pool MSAD3.cariprpc.it iburst
          pool MSAD4.cariprpc.it iburst
          pool MSAD8.cariprpc.it iburst
          pool MSAD9.cariprpc.it iburst
          driftfile /var/lib/chrony/drift
          makestep 1.0 3
          rtcsync
          logdir /var/log/chrony
```

Convertire il file ntp-master.bu in MachineConfig

```
Python

[root@GRPI-OCP-PV00 ntp]# /root/butane ntp-master.bu -o ./ntp-master-conf.yaml
```

Convertire il file ntp-worker.bu in MachineConfig

```
Python

[root@GRPI-OCP-PV00 ntp]# /root/butane ntp-worker.bu -o ./ntp-worker-conf.yaml
```

A questo punto sono state create le policy con il comando seguente:

```
Python
[root@GRPI-OCP-PV00 ntp]# oc kustomize --enable-alpha-plugins=true . | oc apply -f
-
```

IMPORTANTE: la configurazione del servizio *chronyd* prevede il riavvio di tutti i nodi del cluster in maniera *rolling*.
Attendere che tutti i nodi vengano riavviati prima di passare allo step successivo.

A seguito del riavvio è possibile verificare se il MachineConfig è stato applicato correttamente entrando in ssh/debug su un nodo e aprendo il file di configuzione di chronyd

```
Python
[root@GRPI-OCP-PV00 ntp]# ssh <ip node> -l core
```

```
Python
[root@GRPI-OCP-PV00 ntp]# cat /etc/chrony.conf
```

2.3.2. Definizione nodi infrastrutturali

All'interno della directory **/root/policy-generator/acm-hub/ocp-acm-ac/infra-nodes** sono stati definiti i template necessari a:

- Eseguire il deploy dei nodi infrastrutturali
- Creare il *machineconfigpool* per i nodi infrastrutturali

```
Python
[root@GRPI-OCP-PV00]# cd
/root/cluster-acm-policy-generator/acm-hub/ocp-acmac/infra-nodes
```

```
Python
[root@GRPI-OCP-PV00 infra-nodes]# ll
/root/cluster-acm-policy-generator/acm-hub/ocp-acmac/infra-nodes

-rw-r--r--. acm-hub-infra-machineset.yaml
-rw-r--r--. infra-nodes-conf.yaml
-rw-r--r--. kustomization.yaml
-rw-r--r--. mcp-infra.yaml
```

```
Python
[root@GRPI-OCP-PV00 ntp]#  cat kustomization.yaml

generators:
  - infra-nodes-conf.yaml
```

```
Python
[root@GRPI-OCP-PV00 infra-nodes]#  cat infra-nodes-conf.yaml

apiVersion: policy.open-cluster-management.io/v1
kind: PolicyGenerator
metadata:
  name: generator-infra-node-conf-ocp
placementBindingDefaults:
  name: placement-binding-infra-node-conf-ocp
policyDefaults:
  namespace: acm-hub-policy
  placement:
```

```
    clusterSelectors:
      name: local-cluster
      datacenter: ACILIA
  complianceType: musthave
  remediationAction: enforce
  severity: high
policies:
  - name: policy-infra-node-machineset-ocp
    manifests:
      - path: acm-hub-infra-machineset.yaml
  - name: policy-infra-node-mcp-ocp
    manifests:
      - path: mcp-infra.yaml
```

Python

```
[root@GRPI-OCP-PV00 infra-nodes]# cat acm-hub-infra-machineset.yaml


apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: ocp-acmac-btxxb
  name: ocp-acmac-btxxb-infra-0
  namespace: openshift-machine-api
spec:
  replicas: 3
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: ocp-acmac-btxxb
      machine.openshift.io/cluster-api-machineset: ocp-acmac-btxxb-infra-0
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: ocp-acmac-btxxb
        machine.openshift.io/cluster-api-machine-role: worker
        machine.openshift.io/cluster-api-machine-type: worker
        machine.openshift.io/cluster-api-machineset: ocp-acmac-btxxb-infra-0
    spec:
      lifecycleHooks: {}
      metadata:
        labels:
          node-role.kubernetes.io/infra: ""
      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_3_MGMT_AC
      numCPUs: 16
      numCoresPerSocket: 2
      snapshot: ""
      template: ocp-acmac-btxxb-rhcos-generated-region-generated-zone
      userDataSecret:
        name: worker-user-data
      workspace:
        datacenter: ACILIA
        datastore: /ACILIA/datastore/ESX-OCP-PROD-AC-0000
        folder: /ACILIA/vm/PRODUZIONE/OCP
        resourcePool: /ACILIA/host/OCP_PROD//Resources
        server: ac-cags-vcsa001.cariprpc.it
```

Python

```
[root@GRPI-OCP-PV00 infra-nodes]# cat 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
```

Per creare le policy è stato eseguito il seguente comando:

```
Python
[root@GRPI-OCP-PV00 infra-nodes]# oc kustomize --enable-alpha-plugins=true . | oc apply -f -
```

A questo punto è possibile verificare la creazione del *machineset* e del *machineconfigpool*.

```
Python
[root@GRPI-OCP-PV00 infra-nodes]# oc get machinesets -A

KKK

[root@GRPI-OCP-PV00 infra-nodes]# oc get mcp

KKK
```

IMPORTANTE: **Per aumentare la numerosità dei nodi infrastrutturali, andare in edit sul template della policy *acm-hub-infra-machineset.yaml* modificando il valore "replicas" e rilanciare il comando di create della policy, questo aggiornerà la policy con il nuovo valore.**

A questo punto è possibile eliminare i nodi "worker" dal cluster **OCP-ACMAC** eseguendo i seguenti comandi:

Scalare a "0" il MachineSet dei nodi Worker

```
Python
[root@GRPI-OCP-PV00 infra-nodes]# oc scale --replicas=0 machineset ocp-acmac-btxxb-worker-0
```

Eliminare definitivamente il MachineSet

```
Python

[root@GRPI-OCP-PV00 infra-nodes]# oc delete machineset ocp-acmac-btxxb-worker-0
```

N.B: I nodi verranno automaticamente prima Drenati e poi cancellati sia dal cluster Openshift sia da Vmware

2.3.3. Autenticazione tramite LDAP

All'interno della directory "**/root/cluster-acm-policy-generator/acm-hub/all-cluster/oauth"** sono stati definiti i template necessari a:

- Creare la *configmap* contenente la CA per l'utilizzo del protocollo ldaps nella comunicazione con il server LDAP
- Creare la secret contenente l'utenza per eseguire il bind all'LDAP e la password
- Configurare come metodo di autenticazione sul cluster, l'LDAP di Crédit Agricole

```
Python
[root@GRPI-OCP-PV00 oauth]# ll
/root/ocp-acm/cluster-acm-policy-generator/all-cluster/oauth

-rw-r--r--. auth-conf.yaml
-rw-r--r--. bind-secret.yaml
-rw-r--r--. kustomization.yaml
-rw-r--r--. cm-ca.yaml
-rw-r--r--. oauth.yaml
```

```
Python
[root@GRPI-OCP-PV00 oauth]# cat kustomization.yaml

generators:
  - auth-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 oauth ]#  cat auth-conf.yaml

apiVersion: policy.open-cluster-management.io/v1
kind: PolicyGenerator
metadata:
  name: generator-auth-conf-ocp
placementBindingDefaults:
  name: placement-binding-auth-conf-ocp
policyDefaults:
  namespace: acm-hub-policy
  placement:
    clusterSelectors:
      name: local-cluster
  complianceType: musthave
  remediationAction: enforce
  severity: high
policies:
  - name: policy-auth-conf-ocp
    manifests:
      - path: oauth.yaml
  - name: policy-auth-cm-ca-ocp
    manifests:
      - path: cm-ca.yaml
  - name: policy-auth-bind-secret-ocp
    manifests:
      - path: bind-secret.yaml
```

Python

```
[root@GRPI-OCP-PV00 oauth]# cat bind-secret.yaml

apiVersion: v1
data:
  bindPassword: xxxxxxx
kind: Secret
metadata:
  name: ldap-secret
  namespace: openshift-config
type: Opaque
```

Creare la ConfigMap contenente la Certificate Authority del server LDAP

Attenzione: Se dovesse cambiare va recreate questa ConfigMap

Python

```
[root@GRPI-OCP-PV00 oauth]# cat cm-ca.yaml

apiVersion: v1
data:
  ca.crt: |+
    -----BEGIN CERTIFICATE-----
    MIIFKjCCAxKgAwIBAgIQOLXqjbUkq5dN85ulGnvLPTANBgkqhkiG9w0BAQsFADAm
    MSQwIgYDVQQDExtDcmVkaXRBZ3JpY29sZUl0YWxpYVJDQS1QQVIwHhcNMTcwNTAy
    MTU0NDMxWhcNMzcwNTAyMTU1NDMwWjAmMSQwIgYDVQQDExtDcmVkaXRBZ3JpY29s
    ZUl0YWxpYVJDQS1QQVIwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCh
    6zrpKWzle9HixD8Awnq0TnDC3tiDcv009WuWT5qb0PylqHmhjHy06vywNxV12OlV
    +GqWOfvxfJrf3+nWLir7nLxmf05732stWnK2ZK4hS4zaLkG/vt4IKZqkQSal3i1/
    /Ad1Pps8KbDXdgxZME1AxkBM6EbNU1RGoxjT/0xddbfJzZ7Ol5k4rsalM5vqIKfd
    bFBmyrtC7//YabRiUqYi19ulFFCXb4Wf4nu0rMwRhKynPrm+TooiSvDIyb0qEzIS
    3nOxn0ZjvUuL78AyikGWf70ay6tBol2OJTLCWc30tQPNK2CGznjdk45u24bV/X1F
    pGDq8XbdDDP8jnIMX/S/d4ABKcjOmL/cV1oNm5SfrI+E53EXyqW/rJnx40csWKwj
    nFoItg8KUNEC9cgRR/7u/4OVHIiX065mqKef1HNHQGeNlqFPKEMqdWXDYL1jw380
    6g9VF4Wxq40JRN2QyzW++GpMAK88WfsxTbXx5GJ8mA1G/Zm/glPFniZGMDlx1IxS
    jfZ/mXPTDdmact0PwCBT45P7EWcbdosFrnHf5qCo6z8QAUMJv0zlSN6orPOAWald
    fZmpRfhyoQh+DeQ+IoTZ27E/mLm7lLkX1Ixj7Y+sQaAdQ3AjBtHyYgzmPYiI0TJO
    3h63mZWrHXu22tqlTufSmZnQvHxGgiQ14Ruqv0NQIQIDAQABo1QwUjALBgNVHQ8E
    BAMCAYYwEgYDVR0TAQH/BAgwBgEB/wIBATAdBgNVHQ4EFgQUN1a2wutAq3+PW3n2
    AFnJxV7iTUEwEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQELBQADggIBAB7t
    opTXqZhxm+3Fkg3vRoVozBqD1cPZ/NNFE70yKe+vsVXYg+QLfXZE24Uo0CYp/oRZ
    9fJ2t80wIQ0KU2RPZq9Bpi6H37vkV5b6UI55SQZaLwCJfNHkouMqWVH6InfLVF2K
    ARuPTEl33CvDN6sB+PT9IgUZtgpdjj/cCcMZS7v8LmDLFJlFmosEPnu2nTnadfVg
    knlQ//cTiWHWM8ELyK3VnEDXHQqrFWSvjbd8trVi6pYP/aV07a1GWRctryJZGms7
    yMVpW3q4dDK+kd0CerXurdsKuMJkzrcsqD73iPpZiGgilsj2N3iuBm99vwd93HVl
    ha1Lvr1ZCkCL1f2J7iffxk9Wrm1GauFaO25ZydJ5UXu4ihgbnk6AYSK9h19IRc1r
    PmLIghzpqLRXJpZjUa1b6Hytl0jookPbHPb0M6cCK35+L+fdetGXiLSZn4Y1vtqe
    4rjHKa68Lkk4S+ljwz5DvbouCcHAgvOxTkX0M5fkClKS1E7gp2Eu3F8o31dmBlci
    0kZQvFwHFdSNj3zIkxtoRE3TQD2I5CTzMkiURBD264ISjdwc8uKNHadILn1GyaXJ
    lbNvnKb6w7hrcXCqHnTvl1mXu0MlX5zF+h+5mrpGezfEJuK7AlWyj/NsiMVjrVb2
    TaSm5to8Jz08SkHyz2J5baoH0yx0g+AJKxRtnwqt
    -----END CERTIFICATE-----
    -----BEGIN CERTIFICATE-----
    MIIILDCCBhSgAwIBAgITKgAAAANLM56OzEXAcwAAAAAAAzANBgkqhkiG9w0BAQsF
    ADAmMSQwIgYDVQQDExtDcmVkaXRBZ3JpY29sZUl0YWxpYVJDQS1QQVIwHhcNMTcw
    NTAzMDkyODU5WhcNMjcwNTAzMDkzODU5WjBXMRIwEAYKCZImiZPyLGQBGRYCaXQx
    GzAZBgoJkiaJk/IsZAEZFgtjYXJpcHJwY3BhcjEkMCIGA1UEAxMbQ3JlZGl0QWdy
    aWNvbGVJdGFsaWFTQ0EtUEFSMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKC
    AgEAqtmT+HRBlIEG66OzYTzRENNO4Y2LW2CYyGpMIMHzpoAgTtRHQC1JbcKHE/+3
    gyijkipPkbuYbrLea3zMbSgsJHwVasflXK+NufVE1412CFBoLBUUAEivh2zpBkmM
    gOpYo/SvjGwCUuN7vVbraMGbflvDi27iKfMZgzK7hMNNO/ew+MNMqIVq/p95UC3L
    Fx2TJtF6+pUEL1Tm06sWTniLk9PhL0s2re17U/mhz+QrGLsbqom4Zi+LrKB84TX0
    Lis2iSCIIUT4piRccZDh3BF2TOkYrVlPFv1H3+bOpilzZVsSogzWy7I9sP/kDcqv
    K3y2cLHRsckeQcuempJUj5bX+l6F9uDw80gMoBGRi/y2SF7doWQvoHuE2TTLRr0W
    GAbGu2yy5GSO9i5l5i7sdouwi0MEhaYIsEvif68Q3Ja99PLjx1LvDcuIUKBu7hgm
```

```
    uXuXkhMFUVjh8PdSIINpzG8wSPILpQC5uH3n6rGBCEBkqRjdmmLWADdgseU6ESMp
    wT3ViDxSAg3zvWMF1v/H6hnOuBspnDGnUEbuY1JQjQP/aWn9+lfd8PGLem5S5DKL
    Gb4vvWsw36tgoJpFQ6TqyV70z/hLMy8GFeWuOn8aU6NFAyzn56sPGY05KLwjDG+i
    XuauPrl9pIvskvsJ69AkawbS6c2+fk1yBJw97g75EAuJsk8CAwEAAaOCAyAwggMc
    MBIGCSsGAQQBgjcVAQQFAgMBAAEwIwYJKwYBBAGCNxUCBBYEFFnklte59RhxVuJM
    auvEP4SuWNcHMB0GA1UdDgQWBBShy+zN34OTBRtsC41QwKHlT3KY5zAZBgkrBgEE
    AYI3FAIEDB4KAFMAdQBiAEMAQTALBgNVHQ8EBAMCAYYwEgYDVR0TAQH/BAgwBgEB
    /wIBADAfBgNVHSMEGDAWgBQ3VrbC60Crf49befYAWcnFXuJNQTCCAS0GA1UdHwSC
    ASQwggEgMIIBHKCCARigggEUhoHLbGRhcDovLy9DTj1DcmVkaXRBZ3JpY29sZUl0
    YWxpYVJDQS1QQVIsQ049R1JQSS1DQUEtSFYzMSxDTj1DRFAsQ049UHVibGljJTIw
    S2V5JTIwU2VydmljZXMsQ049U2VydmljZXMsQ049Q29uZmlndXJhdGlvbixEQz1D
    QVJJUFJQQ1BBUixEQz1pdD9jZXJ0aWZpY2F0ZVJldm9jYXRpb25MaXN0P2Jhc2U/
    b2JqZWN0Q2xhc3M9Y1JMRGlzdHJpYnV0aW9uUG9pbnSGRGh0dHA6Ly9wa2kuY2Fy
    aXBycGNwYXIuaXQvQ2VydEVucm9sbC9DcmVkaXRBZ3JpY29sZUl0YWxpYVJDQS1Q
    QVIuY3JsMIIBMgYIKwYBBQUHAQEEggEkMIIBIDCBvQYIKwYBBQUHMAKGgbBsZGFw
    Oi8vL0NOPUNyZWRpdEFncmljb2xlSXRhbGlhUkNBLVBBUixDTj1BSUEsQ049UHVi
    bGljJTIwS2V5JTIwU2VydmljZXMsQ049U2VydmljZXMsQ049Q29uZmlndXJhdGlv
    bixEQz1DQVJJUFJQQ1BBUixEQz1pdD9jQUNlcnRpZmljYXRlP2Jhc2U/b2JqZWN0
    Q2xhc3M9Y2VydGlmaWNhdGlvbkF1dGhvcml0eTBeBggrBgEFBQcwAoZSaHR0cDov
    L3BraS5jYXJpcHJwY3Bhci5pdC9DZXJ0RW5yb2xsL0dSUEktQ0FBLUhWMzFfQ3Jl
    ZGl0QWdyaWNvbGVJdGFsaWFSQ0EtUEFSLmNydDANBgkqhkiG9w0BAQsFAAOCAgEA
    XbBUDlc4FeCJGdJeWWDvgK2brxO9VshgjgZV9GGmmFlLR/TKim0VkrRAjfSvxf1x
    nto8bHZ2ivxBr0RSBVOXEvVYycMIkMxpbGbKAIkN2PnuMO5FCudNYpMA4P0HcQch
    wiGQlonj7e2/Azvtyc5ML4xLubb6JQNt1L/76dCdheGmdYj3YNkrIzaDkMe3jZmW
    TrVI+hOLOL9XuOnNzQkhZ0INQ/QQmUGQ6xdaM1z5MVkOvTPpkzhD8s6Vt/nkodOA
    J9gQEDlhO46tl67QqplsA8jX3wgRqvffZ3E9ZSx3t1dDsnslhsZEoDHVALTMcyfU
    EMEsGtRlkWPrUcEPzBnn9mTegaoEAby53OAVaW5FeT2iGSlCIhukiu70Mvf8MIAA
    /p3kDl2J3OtUJ/THjg8952JCa8WBpvON25cV4QU0PhfJmnBfoVHYmpAntcXcECS1
    OHJnkD0pRgqKz+GcW7mIVWrygaXwH+nXrJw22ympf+s2h3xvjPqFCy5PxDjZIyaP
    OD2S8PIWHfQZTztYhhsTl675U6oGO3T5/BRxQ+s+AKtjUpQ24Wz52tdulD9Ya1Xc
    f6Msid7qH9Nv9lT5sjyy+bNdla2YFvXcjEIh7g8GVqRWszkbbb2irU0f5PbjiApV
    Yv4sy+JZ53ZzY07YmIHBlvFPlm4194MNTBgbpd+zBHE=
    -----END CERTIFICATE-----
kind: ConfigMap
metadata:
  name: ca-config-map
  namespace: openshift-config
```

Python

```
[root@GRPI-OCP-PV00 oauth]# cat oauth.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_4462_ocp_ldap,OU=OU-UTENTI-SERVIZI,OU=OU-UTENTI,DC=cariprpcpar,DC=it
      bindPassword:
        name: ldap-secret
      ca:
        name: ca-config-map
      insecure: false
      url:
ldaps://msad0par.cariprpcpar.it/DC=cariprpcpar,DC=it?sAMAccountName?sub?(&(objectCl
ass=user)(|(memberOf=CN=Users,DC=cariprpcpar,DC=it)(memberOf=CN=GU_DTR_USER,CN=User
s,DC=cariprpcpar,DC=it)(memberOf=CN=GU_OCP_ADMIN,CN=Users,DC=cariprpcpar,DC=it)(mem
berOf=CN=GU_OCP_USER,CN=Users,DC=cariprpcpar,DC=it)))
    mappingMethod: claim
    name: ldap
    type: LDAP
```

Per configurare l'operator dell'autenticazione è necessario eseguire il seguente comando:

```
Python
[root@GRPI-OCP-PV00 oauth]# oc kustomize --enable-alpha-plugins=true . | oc apply
-f -
```

Attendere il riavvio dei pod dell'autenticazione prima di testare la login tramite LDAP.

Verificare il riavvio corretto dei Pod di autenticazione:

```
Python
[root@GRPI-OCP-PV00 oauth]# oc get po -n openshift-authentication
```

2.3.4. Sync gruppi di utenti tra alberatura LDAP e OCP

All'interno della directory **/root/cluster-acm-policy-generator/acm-hub/all-cluster/group-sync-operator** sono stati definiti i template necessari a:

- Creare il namespace che ospita il group-sync-operator
- Installare il group-sync-operator
- Creare la *configmap* contenente la CA per l'utilizzo del protocollo ldaps nella comunicazione con il server LDAP: il campo ca.crt contiene il *base64encode* della CA.
- Creare la secret contenente l'utenza per eseguire il bind all'LDAP e la password
- Configurare il group sync operator
- Associare il ruolo cluster-admin agli utenti appartenenti al gruppo CN=GU_OCP_ADMIN,CN=Users,DC=cariprpc,DC=it

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# ll
/root/ocp-acm/cluster-acm-policy-generator/acm-hub/all-cluster/group-sync-operator

-rw-r--r--. group-sync-conf.yaml
-rw-r--r--. ldap-ca-bundle-group-sync.yaml
-rw-r--r--. ldap-creds-group-sync.yaml
-rw-r--r--. ldap-groupsync.yaml
-rw-r--r--. namespace.yaml
-rw-r--r--. operatorgroup.yaml
-rw-r--r--. role-bindig.yaml
-rw-r--r--. subscription.yaml
-rw-r--r--. kustomization.yaml
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat kustomization.yaml
generators:
  - group-sync-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat group-sync-conf.yaml

apiVersion: policy.open-cluster-management.io/v1
kind: PolicyGenerator
metadata:
  name: generator-group-sync-conf-ocp
```

```
placementBindingDefaults:
  name: placement-binding-group-sync-conf-ocp
policyDefaults:
  namespace: acm-hub-policy
  placement:
    clusterSelectors:
      name: local-cluster
  complianceType: musthave
  remediationAction: enforce
  severity: high
policies:
  - name: policy-group-sync-namespace-ocp
    manifests:
      - path: namespace.yaml
  - name: policy-group-sync-operatorgroup-ocp
    manifests:
      - path: operatorgroup.yaml
  - name: policy-group-sync-subscription-ocp
    manifests:
      - path: subscription.yaml
  - name: policy-group-sync-ldap-groupsync-ocp
    manifests:
      - path: ldap-groupsync.yaml
  - name: policy-group-sync-ca-secret-ocp
    manifests:
      - path: ldap-ca-bundle-group-sync.yaml
  - name: policy-group-sync-bind-secret-ocp
    manifests:
      - path: ldap-creds-group-sync.yaml
  - name: policy-group-sync-admin-ocp
    manifests:
      - path: role-bindig.yaml
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat namespace.yaml

apiVersion: v1
kind: Namespace
metadata:
  name: group-sync-operator
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]#  cat operatorgroup.yaml
```

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: group-sync-operator
  namespace: group-sync-operator
spec:
  targetNamespaces:
  - group-sync-operator
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat subscription.yaml

apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: group-sync-operator
  namespace: group-sync-operator
spec:
  channel: alpha
  installPlanApproval: Manual
  name: group-sync-operator
  source: community-operators
  sourceNamespace: openshift-marketplace
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]#  cat ldap-groupsync.yaml

apiVersion: redhatcop.redhat.io/v1alpha1
kind: GroupSync
metadata:
  name: ldap-groupsync
  namespace: group-sync-operator
spec:
  providers:
  - ldap:
      activeDirectory:
        groupMembershipAttributes:
        - memberOf
        userNameAttributes:
```

```
        - sAMAccountName
        usersQuery:
          baseDN: DC=cariprpc,DC=it
          derefAliases: never
          filter:
(&(objectClass=user)(|(memberOf=CN=Users,DC=cariprpc,DC=it)(memberOf=CN=GU_DTR_USER
,CN=Users,DC=cariprpc,DC=it)(memberOf=CN=GU_OCP_ADMIN,CN=Users,DC=cariprpc,DC=it)(m
emberOf=CN=GU_OCP_USER,CN=Users,DC=cariprpc,DC=it)))
          pageSize: 0
          scope: sub
      caSecret:
        kind: Secret
        name: ldap-ca-bundle-group-sync
        namespace: group-sync-operator
      credentialsSecret:
        kind: Secret
        name: ldap-creds-group-sync
        namespace: group-sync-operator
      insecure: false
      url: ldaps://msad1.cariprpc.it
      whitelist:
      - CN=Users,DC=cariprpc,DC=it
      - CN=GU_DTR_USER,CN=Users,DC=cariprpc,DC=it
      - CN=GU_OCP_ADMIN,CN=Users,DC=cariprpc,DC=it
      - CN=GU_OCP_USER,CN=Users,DC=cariprpc,DC=it
    name: ldap-group-sync
  schedule: '*/5 * * * *'
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]#  cat ldap-ca-bundle-group-sync.yaml

apiVersion: v1
data:
  ca.crt:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUZ0akNDQTU2Z0F3SUJBZ0lRTFl0ajZpdXQ0NTlPUnR
BV3JvYW40ekFOQmdrcWhraUc5dzBCQVFzRkFEQnMKTVFzd0NRWURWUVFHRXdKSlZERWZNQjBHQTFVRUNoTV
dRM0psWkdsMElFRm5jbWxqYjJ4bElFbDBZV3hwWVRFYQpNQmdHQTFVRUN4TVJVSEpwZG1GMFpTQkhjbTkxY
0NCUVMwa3hJREFlQmdOVkJBTVRGME55WldScGRFRm5jbWxqCmIyeGxTWFJoYkdsaFVrTkJNQjRYRFRFM01E
VXdNekUwTWpZeU9Wb1hEVFEzTURVd016RTBNell5Tmxvd2JERUwKTUFrR0ExVUVCaE1DU1ZReEh6QWRCZ05
WQkFvVEZrTnlaV1JwZENQlozSnBZMjlzWlNCSmRHRnNhV0V4R2pBWQpCZ05WQkFzVEVWQnlhWFpoZEdVZ1
IzSnZkWEFnVUV0Sk1TQXdIZ1lEVlFREV4ZERjbVZrYVhSQlozSnBZMjlzClpVbDBZV3hwWVZKRFFUQ0NBa
Ul3RFFZSktvWklodmNOQVFFQkJRQURnZ0lQQURDQ0Fnb0NnZ0lCQU1wNzl4NnEKdmE2VHdaK0ZYNkRUdTIz
```

YzQyWjdWWmwrdDJxb01HN2NzY0J1VzNJVlRhbjBPdTV6d3hscGlXMnN5Tmx1OUJOMwo0Y2JHYm5CK2poeWc
vN0Z2NmRxbUZoRkpmQVBYM1JSMGpvZDUzS2ZqenZqZmMvSXBvZzExajhtZkFSMWwzTjY2CmFwU1krM0o5dU
5mTmpvT2xWZXUrU0hYVGxYbFVZNVJuNlp6a0E0VEM0TlJYeWY5b0Z3ZkJsem1tWXYvK1pHS0kKOWlUKzd6V
XZWM2Q5bHJCV1QybkN0N3A2eWhENG5ESUVzY0I4d0dvQ3ZScmtiTjJFdDJNVll2UFpYaHFhYVMxRwpLdi9M
S0ZlOU9LUjJyS1VHckxkbzB1aWcyQzVuQXBlTXF4K31MNVhORGVUc1NwNEpjS0JNZmxMTVZ6UHFBSW4xClV
IUnc2cDVXdmtsb3ROaTFycDIvUkZucDFYenh3YkdkVTZwM2VXYVBtckMvZ0NnT1hqcW56Q1dwZDZiVW0xRT
YKclZFNkNON0tSU3FadWptTlZDb1BKQmtXaFVJeTdOSlJLT0N0bCtPVUxoNTVGV2VKbnJoS0ZjMWxOVVRWb
0tQNwpYVys2WE9pNGVyc1hHUUN5RTd3bE1ZenJMUDFSdFhrK2dWanZyUHhCMklsWHpMalZMRnZvN1FoL0RO
UHhaNEFJCng5WHVYa1IzMElqWnhkR2VlM0VVd2p0dmFoclg2U1BVV2w4NzIxY2ovTUgxTkd2Q2hHZTZ1c29
oa0ZCeXdKc24KS1R2a0U4U2prRDQ0R2dxZW1pRlNVMHBrN0Z0S2RSZ3RYYTFXa1VwWWwwMzhxUHpTaHdJbW
01SDRQQVpqU2k3cgowOHVCZ0JLdG9KcVJHL0Y1dzFoZkdwbDFxQnJKN3FPZXdQUVBBZ01CQUFHalZEQlNNQ
XNHQTFVZER3UUVBd0lCCmhqQVNCZ05WSFJNQkFmOEVDREFHQVFIL0FnRUJNQjBHQTFVZERnUVdCQlFYNlJ1
anZDNVBLNTd2ejBxTmZCQ3YKbkd4KzB6QVFCZ2tyQmdFRUFZSTNGUUVFQXdJQkFEQU5CZ2txaGtpRzl3MEJ
BUXNGQUFPQ0FnRUFJeGhFbUUrdQpjYjZ0UzRPdENHUjhpb3VGK2xjYlFIRnpOUVlhUWxwRUE5c3FSekl2MW
Uvc2o5WnBFNDRsU1ZSakdMWVhnOHViCkVjVnVwWGxMeG5GVWt6dVJNVjhEc0RxZkR5T3NuM2hFS3dlOFJzc
nQzQUFTa1JycDFnOHZhcTlScGViTVM0WWgKNnRsa1BLVWJQQ0NDZXdjaER3dWIzQXpCU3EzRWkzOEk0Ui9I
VjRyRHpZK3N4SHRkRDV4UmNuaUdtWTJneTY5VwpLOHdoTUZEQ1RzdXJVYWdBNllluRURrQlRiRk0vTWlYUkh
UczBLUWE1R1I0cGVhY1FVREsrckJUWWloN3YvUG5yCnRMVFdhRnYyaTBJUVRGcWd4N0RJb1JWUzlIOFNLRE
pkamovbktSWmk0d3Y2b083TEUydG1qa2U3MzIzeElDejMKQjRaQmNKQVo2aUZ4NFhNTXJRNlowNE5IRUdHY
WtGQVV4Z2h3OGVJc3oyb2lKKzZRK2oxNmF4eEFldGprd1RMQgprY2lSanJ0N0FKYkFieEF2SWg5TFNOU201
K1NzQ2VhTDY0UjV6U2psMmV5NVBzT000dzdaN3J1dTU1M3BIcDEvCkkzdzEvZER2UTZteitRb0lHV0REbTd
zMDk1bHRSZEF6TlNuK00vaGVEZWxMSVdqZnhGMm4wNTJVQ0lOeERLL1MKZ1RLUmlnbDQ0NkxRL0cyc0tDWS
tYVFo0Vjk0aWp1Mk95R0pDdk5vTEtHMUdoS0hsWGJMQ3FpU3ZQRnE0bE9ZMwo1MTNBOXFEcnBXUjBEanBDb
TZoRjVaNXhSUHFvdnZhSmhtbStTd1oyMXpQMHY4M3lOS0tTanVub0ROUTZWakw3CkRSVWpMZGhVR2hhaW10
SGFnSnFkWlV4dlRsNUhBQit0N1F3PQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCi0tLS0tQkVHSU4gQ0V
SVElGSUNBVEUtLS0tLQpNSUlJVWpDQ0JqcWdBd0lCQWdJVEtnQUFBQUpoelBJNHA0T2YrQUFBQUFBQUFqQU
5CZ2txaGtpRzl3MEJBUXNGCkFEQnNNUXN3Q1FZRFZRUUdFd0pKVkRFZk1CMEdBMVVFQ2hNV1EzSmxaR2wwS
UVGbmNtbGpiMnhsSUVsMFlXeHAKWVRFYU1CZ0dBMVVFQ3hNUlVISnBkbUYwWlNCSGNtOTFjQ0JRUzBreElE
QWVCZ05WQkFNVEYwTnlaV1JwZEVGbgpjbWxqYjJ4bFNYUmhiR2xoVWtOQk1CNFhEVEUzTURVd016RTFNRGN
5TjFvWERUSTVNRFV3TXpFMU1UY3lOMW93CmJERUxNQWtHQTFVRUJoTUNTVlF4SHpBZEJnTlZCQW9URmtOeV
pXUnBkQ0JCWjNKcFkyOXNaU0JKZEdGc2FXRXgKR2pBWUJnTlZCQXNURVZCeWFYWmhkR1VnUjNKdmRYQWdVR
XRKTVNBd0hnWURWUVFERXhkRGNtVmthWFJCWjNKcApZMjlzWlVsMFlXeHBZVk5EUVRDQ0FpSXdEUVlKS29a
SWh2Y05BUUVCQlFBRGdnSVBBRENDQWdvQ2dnSUJBTEpZCjM3dG9BZFBYM2lTVnZxTHA3RHFqQjBEWXllVWp
BcVBnVnh0V2JGbU9CelBpMHo4S3Q1bUs4MmxIUi9Db3FxTlAKUU9MWjZIbUY4dTlqU3ljNGlaRjZkU2V2OF
YzQkxDVWEzWmZmYjlKM2hIb2cwMWdhcnFIbUlpaytOVjlRNTl1WApIZjFCOTRmaUJ1UmFhdVVKTTVHUG05O
Dl5dnBqT0NLa2ppaUxlQ2dBSG5EQUtnZ3c3N2twcEVUYzMxUklNWklkCmR1TzZ4K2FoMmZHeUE2NEZOVkJp
UDdVa0RXK2djNkhiUEx4UDA2QTRuZmc1MnI5a2NSL0x5UUxQVXM3WmxYLzcKNjR2d2MwNUtwZ1RHU3hYTml
JdFZjUnRMSm9wWFlBYlRkZ0ZHUFl6L05JQkorZTRwMHNoY29VRnNDREhUSGZpegozcEFXODQxb3pyZWpRQn
ZQekYxcXJCZEQ2OFFXKzBGODUrZXV1QStITUtEYXR1a0g4Y31URWJUMzVpalFmTGdlCkE0YjA3TGh3R2ljd
GN2RU1qWE15c25kV29yK2JLS2NTMG9FSVBWbGpad0t4bHUzTGZkMHRVOTZIYVVDQktYWmcKUkVwd1A2SWdQ
aDhhVFFLOXBCN0h2cW1CTmp5elkvMU9UVkFJYld1N1N6WWN0RWx2cmFCYjRIRzVVRzEvVEhyNgprMjVzUk8
yNlpjVEp6T3RXNWVlRnM1T0haMER2Qk5oYk1aYnJEMjlJWUxOaTdNdU5YcWpZOEpWRHFnSEtEeGVxCnV3M1
dmZHdFS2NGd1YySnVXNVN6RVZVREhzY3kzQTZXcm1wd3pFTlV6ejVvOXhKNklGc0lxQUE1TDkwdUlzanMKU
ytJOUFSc2ZqV29GdDNJNXUxbGlCMUtPaXZhREVFbnFXbnJKYXdoMUFnTUJBQUdqZ2dMck1JSUM1ekFRQmdr
cgpCZ0VFQVlJM0ZRRUVBd0lCQURBZEJnTlZIUTRFRmdRVTRBclpuZEpodjZQeVI5Z3RFcU5kNFlyNHRIc3d
HUVlKCkt3WUJCQUdDTnhRQ0JBd2VDZ0JUQUhVQVlnQkRBRUV3Q3dZRFZSMFBCQVFEQWdHR01CSUdBMVVkRX
dFQi93UUkKTUFZQkFmOENBUUF3SHdZRFZSMGpCQmd3Rm9BVUYra1hvN3d1VHl1ZTc4OUtqWHdRcjV4c2Z0T
XdnZ0VtQmdOVgpIUjhFZ2dFZE1JSUJHVENDQVJXZ2dnRVJvSUlCRFlhQnhHeGtZWEE2THk4dlEwNDlRM0ps
WkdsMFFXZHlhV052CmJHVkpkR0ZzYVdGU1EwRXNRMDQ5UjFKUVNTMURRVUV0VUZZek1TeERUajFEUkZBc1E
wNDlVSFZpYkdsakpUSXcKUzJWNUpUSXdVMlZ5ZG1salpYTXNRMDQ5VTJWeWRtbGpaWE1zUTA0OVEyOXVabW
xuZFhKaGRHbHZiaXhFUXoxRApVkpKVUZKUVF5eEVRejFwZEQ5alpYSjBhV1pwWTJGMFpWSmxkbTlqWVhSc
GIyNU1hWE4wUDJKaGMyVS9iMkpxClpXTjBRMnhoYzNNOVkxSk1SR2x6ZEhKcFluVjBhVzl1VUc5cGJuU0dS
R2gwZEhBNkx5OXdhMmt1WTNKbFpHbDAKTFdGbmNtbGpiMnhsTG1sMEwwTmxjblJGYm5KdmJHd3ZRM0psWkd
sMFFXZHlhV052YkdWSmRHRnNhV0ZTUTBFdQpZM0pzTU1JQkt3WUlLd1lCQlFVSEFRRUVnZ0VkTUlJQkdUQ0
J0Z1lJS3dZQkJRVUhNQUtHZ2Fsc1pHRndPaTh2CkwwTk9QVU55WldScGRFRm5jbWxqYjJ4bFNYUmhiR2xoV
WtOQkxFTk9QVUZKUVN4RFRqMVFkV0pzYVdNbE1qQkwKWlhrbE1qQlRaWEoyYVdObGN5eERUajFUWlhKMmFX
TmxjeXhEVGoxRGIyNW1hV2QxY21GMGFXOXVMRVJEUFVOQgpVa2xRVWxCRExFUkRQV2wwUDJOQlEyVnlkR2x

```
tYVdOaGRHVS9ZbUZ6WlQ5dlltcGxZM1JEYkdGemN6MWpaWEowCmFXWnBZMkYwYVc5dVFYVjBhRzl5YVhSNU
1GNEdDQ3NHQVFVRkJ6QUNobEpvZEhSd09pOHZjR3RwTG1OeVpXUnAKZEMxaFozSnBZMjlzWlM1cGRDOURaW
EowUlc1eWIyeHNMMGRTVUVrdFEwRkJMVkJXTXpGZlEzSmxaR2wwUVdkeQphV052YkdWSmRHRnNhV0ZTUTBF
dVkzSjBNQTBHQ1NxR1NJYjNEUUVCQ3dVQUE0SUNBUUM5UTd3cGFwRnQ0Wk9XCmJaMnZaN0pjUGMwYUtRZnl
oMVd1clBYTk1JaGdxTkdackVXdHYrVGRtQ3Z5VHk3eW03Z0JrM1ZyQVhtWVhYdDUKZ3FlRnl0dHZCdEFMOD
Nnd05BWmNoSmxqWVFsYXJPek1WSXNnTVluUGFjK29xeTI1dVI1MHlCenI1ek8rZ1BjYgo2TzNLQzRLL200e
TdRMFhzSDNNa2VMMXpmRkgwYkFNMnFMYnVvcE5kY3hTWGw2bVFES2dENUJBV0xGUDFGbGtYCkxZLzdmQzh6
R0o4SjlnZVJrOFZGR3VQL1N2RnZvcW5DUHBFYU42RmVLbWtBaWxZTGlwZjNkUjU5YUJ5SUNYWksKWGhLTld
VWG81dTZxdW1ZZlNDbW5SbWVCZXRDTnNiRGt4aVd3UXZOd1pXY1diMG45elFNUzJCRnRTS0loMGpJVApzWD
dkTUtQRnFPQ0I3SHcyRi8zdmh5b3JDeFA1K0d3eDVDS0UwRWdiRkVoVUd3YmNrTHQ3OW05dDVtOWF3bGJhC
mdiVll4ampyNzhpd21icXQwejlXd3RERHhZVldpVDFSSmtmV3E4R3JNLzlIaHVOS2ZUbVZYUVFyaW1uRXNh
T2gKNllVU281N2prZSt1NHFQL25CalowdVVOaGV6WmV4N3hDRGF0aDRKckVEZEZaNFJ2MjRKNWhqWTlFaUF
mbHdQSApwM3V5TjhLNHRMV3VKdmU0N3dhZnQzaEU2WUJHK1V2NTdxbUdRbllESGt1NzVRbmErd1ljOFRJMl
R2c3ZKZWlyCk9PSjlpeHpaYUhoMEZFbkpXVmU0Y0xPM2preEZlb09EM3B1WVE4VXJPVUJwZDVQY3phNVg2a
mJVNE4zdjNXOFgKcHFsQUx0Tmg4WnR6VXlibVVpK1RBbmpvY092akV3PT0KLS0tLS1FTkQgQ0VSVElGSUNB
VEUtLS0tLQo=
kind: Secret
metadata:
  name: ldap-ca-bundle-group-sync
  namespace: group-sync-operator
type: Opaque
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat ldap-creds-group-sync.yaml

apiVersion: v1
data:
  password: TEtlSDBiN1BhZWhvTyRLZzFrdVA=
  username:
Q049Y3BfNDQ2Ml9vY3BfbGRhcCxPVT1PVS1VVEVOVEktU0VSVklaSSxPVT1PVS1VVEVOVEksREM9Y2FyaXB
ycGMsREM9aXQK
kind: Secret
metadata:
  name: ldap-creds-group-sync
  namespace: group-sync-operator
type: Opaque
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]#  cat role-bindig.yaml

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: clusteradmin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
```

```
  name: cluster-admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: CN=GU_OCP_ADMIN,CN=Users,DC=cariprpc,DC=it
```

Per creare questa policy è stato lanciato il comando seguente:

```
Python
[root@grpi-ocp-hv00 group-sync-operator]# oc kustomize --enable-alpha-plugins=true
. | oc apply -f -
```

IMPORTANTE: una volta creata la policy, collegarsi alla console di OCP, ed approvare a mano l'installation-plan dell'operator group-sync-operator.

2.3.5. Installazione ODF

I passi eseguiti sono i seguenti:

- Collegandosi alla console di OCP del cluster OCP-ACMAC, all'interno del tab "Operator Hub", digitare "OpenShift Data Foundation" e cliccare su "install".
- Nella pagina della subscription che si apre, lasciare il default "openshift-storage" come namespace di default per l'installazione.
- Modalità di aggiornamento: selezionare "Automatic".
- Abilitare il plugin della console
- Cliccare su Install

Una volta installato l'operator, selezionarlo e modificare la strategia di aggiornamento della subscription, da "Automatic" a "Manual".

2.3.6. Configurazione ODF

I passi eseguiti sono i seguenti:

- Aprire operator OpenShift Data Foundation
- Creare StorageSystem
- Selezionare “MultiCloud Object Gateway”
- Selezionare la StorageClass di Vmware “thin-csi”
- Lasciare il resto di Default
- Attendere completamento dell’installazione

Una volta completata l’installazione creare un nuovo pvc per testare le nuove StorageClass

2.3.7. Aumento replica IngressController a 3

Dato che i nodi Infra previsti sono con replica 3 è buona norma incrementare anche le repliche del pod Ingress

```
Python
[root@GRPI-OCP-PV00 group-sync-operator]#
oc patch ingresscontroller/default -n openshift-ingress-operator --type=merge -p
'{"spec":{"replicas": 3}}'
```

2.4. Installazione cluster base Rozzano

La piattaforma che ospita il cluster OCP di management è un vSphere in versione 7.0.3 , compatibile con la versione di OCP scelta, la 4.14.2.

Tale cluster è composto dalle seguenti macchine virtuali:

Ruolo	vCPU	RAM	OS disk	Count
Master	16	32	120 GB	3

Infra	16	32	120 GB	3

Come attività preliminari all'installazione sono stati eseguiti i seguenti task:

- Impostazione del proxy sul bastion **GRPI-OCP-PV00**
- Creazione della directory di lavoro **/root/ocp-acmrz**
- Creazione chiave ssh per l'accesso ai nodi OCP **/root/.ssh/ocp-produzione**
- Download client e installer in versione **4.14.2**
- Copia in locale dei certificati del vCenter
- Verifica/Download della Pull Secrets dal Portale RedHat

Creare Directory di lavoro

```
Python
[root@GRPI-OCP-PV00 ~]# mkdir /root/ocp-acmrz
```

```
Python
[root@GRPI-OCP-PV00 ~]#  cd ocp-acmrz
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# mkdir install_dir
```

Impostare i proxy

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# export
https_proxy=http://vip-navproxy-server.cariprpc.it:8080
[root@GRPI-OCP-PV00 ocp-acmrz]# export
http_proxy=http://vip-navproxy-server.cariprpc.it:8080
```

```
[root@GRPI-OCP-PV00 ocp-acmrz]# export
no_proxy=localhost,127.0.0.1,localaddress,.localdomain.com,.cariprpc.it,10.68.0.0/1
4,172.27.0.0/16,10.215.87.0/24
```

Scaricare Client e Installer di Openshift

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# wget
https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.14.2/openshift-i
nstall-linux.tar.gz

[root@GRPI-OCP-PV00 ocp-acmrz]# wget
https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/4.14.2/openshift-c
lient-linux-4.14.2.tar.gz
```

Scompattare il Client e Installer

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# tar zxvf openshift-install-linux.tar.gz

[root@GRPI-OCP-PV00 ocp-acmrz]# chmod +x /root/ocp-acmrz/openshift-install
```

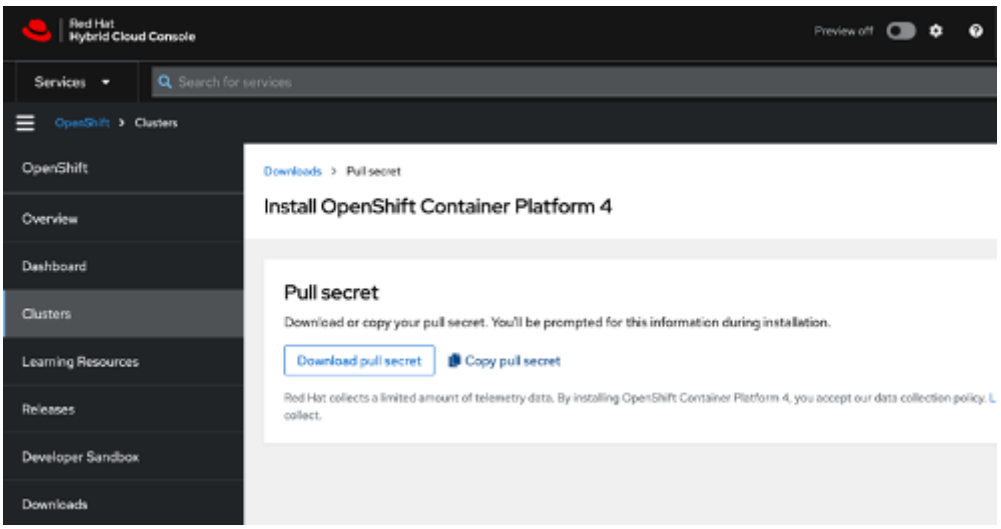
```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# tar zxvf openshift-install-linux.tar.gz
[root@GRPI-OCP-PV00 ocp-acmrz]# chmod +x /usr/bin/oc
[root@GRPI-OCP-PV00 ocp-acmrz]# tar zxvf openshift-client-linux-4.14.2.tar.gz -C
/usr/local/sbin
```

Scaricare e “Trustare” i certificati di Vmware

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz ~]# cd /root

[root@GRPI-OCP-PV00 ocp-acmrz ~]# wget
https://ac-cags-vcsa001.cariprpc.it/certs/download.zip

[root@GRPI-OCP-PV00 ocp-acmrz]# cp /root/certs/lin/*
/etc/pki/ca-trust/source/anchors

[root@GRPI-OCP-PV00 ocp-acmrz]# update-ca-trust
```

ATTENZIONE: Come indicato è stata utilizzata la stessa chiave SSH per entrambi i cluster non è necessario quindi generarla nuovamente.

Per poter installare correttamente OCP il file di configurazione chiamato install-config necessita delle credenziali di accesso al registry di RedHat, tale credenziali è chiamata "pull Secrets" ed è scaricabile al seguente link



A questo punto è stato preparato l'*install-config.yaml* che riportiamo:

```
Python

apiVersion: v1
baseDomain: cariprpc.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,.cariprpc.it,.cariprpcpar.it,.car
iprpccoll.it,172.30.0.0/16,10.19.84.0/22,10.19.87.7,10.19.87.6
compute:
- name: worker
  hyperthreading: Enabled
  platform:
    vsphere:
      cpus: 16
      coresPerSocket: 2
      memoryMB: 32768
      osDisk:
        diskSizeGB: 120
  replicas: 3
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    vsphere:
      cpus: 16
      coresPerSocket: 2
      memoryMB: 32768
      osDisk:
        diskSizeGB: 120
  replicas: 3
metadata:
  name: ocp-acmrz
networking:
  machineNetwork:
  - cidr: 10.19.84.0/22
  networkType: OVNKubernetes
  serviceNetwork:
  - 172.30.0.0/16
platform:
  vsphere:
    apiVIP: 10.19.87.7
    cluster: "OCP_PROD"
    datacenter: "ROZZANO"
    defaultDatastore: ESX-OCP-PROD-RZ-0000
    ingressVIP: 10.19.87.6
    network: "dvpg_3_MGMT_AC"
    password: 'N65sCPnrD$AasJANJThc'
    username: cariprpc\cp_12_vcenter_OCP
    vCenter: rz-cags-vcsa001.cariprpc.it
    folder: "/ROZZANO/vm/PRODUZIONE/OCP"
publish: External
pullSecret:
'{"auths":{"cloud.openshift.com":{"auth":"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc
3NfZTdjYzcwZDhkNjZlNGE1MTgxYWZlOTAzYTNiMTI2NDk6TjZHSFpKUTI4RFExNEgzT05PVlQ2UTZYUU4z
MU40UU9TVzRYTFNEVjdUOElBVDA3V01VTFpVSFBFQzRYTkFSVg==","email":"daniele.bagiotti@cre
dit-agricole.it"},"quay.io":{"auth":"b3BlbnNoaWZ0LXJlbGVhc2UtZGV2K29jbV9hY2Nlc3NfZT
djYzcwZDhkNjZlNGE1MTgxYWZlOTAzYTNiMTI2NDk6TjZHSFpKUTI4RFExNEgzT05PVlQ2UTZYUU4zMU40U
U9TVzRYTFNEVjdUOElBVDA3V01VTFpVSFBFQzRYTkFSVg==","email":"daniele.bagiotti@credit-a
gricole.it"},"registry.connect.redhat.com":{"auth":"fHVoYy1wb29sLTgxNGFiODI0LTgwMzc
tNGJjMy1iMTA2LWUwZDQyMWY5NDU2ZjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSTJNekZpWW
```

```
1Ka09XTXlOVGswWTJaak9UWm1abUkwWVRRMU4yTmhORE00TVNKOS5jdVU0TjA4R3Q5VUh6Snk0d3VXOV9ae
FdTc1FiN0dUYThWRm0yTTdqWFBmSXd3TXZ0YnZGTHVwYno5bTZZX3VXTE91UF9SZk9xSVpldGk1X3oxUnBN
OFVnRDMzUUJ2V2QxbGFxMkJEY0ZUTHlyQ2Z5anVvaGdvZFppZE5SRFgwQXRURFdDLTlOMmRlMTZMUWhaeUt
tYnBaZlFDVVo5TGtDLWdQSk1xVmNTOFY3RzhXM2x0QWkyTm55TGp1dzBDMk5MUjl3M3R4M29tRnhES1Jabm
1kYmFQemdzQ2o4bnNCZXhDYmtYRFVCVWo1YUFjX1Fha3FxNVZZYmtxYTQtNHRZYTljamc3RXEtUy11dW5pW
lo3cldxLUhOdmlYRUpWVVY0M1NkdnlNZ2I0bTQwanFaYVpqOVpLQ1pycktVUdnd0tTS1IteUNwcHhWZWZS
cnB0QmZ4WndrM0ZPNGg1azZuOUdOSE1RazdmRGVnTEJ5WVdmcVdQNnZHSENudW1LRVFBLWczcXBPZENrWlp
tV2VRQlpHaExHNkhpS0luRHVIOUxKb2hhSm5BcTBqVVhsVHdGYWZ4OUtoVXhUemM1cDhBUmQwZGdEd2tUZj
JaVWZKM2czVW1QUjNHaTRmY2xTaHc5RW9UbXlKXzYzdHpSV0pJd053X0VSOE9QSHVnNFNwM2pfZ2lPaXNCS
mI2TU5ybkJ3WmFXWElDNjdCbnR2REJVTXgyZkdYbDJ2cDZEamdpWjZxU2Z5blRwWXFpaFBGSWplbldQd0Vq
UFl5b01nQTF2NVJoanppT1ZCemZXRTVKY1pOdC1VaU5GckdaMmV0bXdQUkpNaThJODJ2YWx3SjlxYkNZT21
fQ2xsRWI4TEF0aWJDTGtyYUVPRmY2cmpyMGRSUXZ1VlRIb3BJTQ==","email":"daniele.bagiotti@cr
edit-agricole.it"},"registry.redhat.io":{"auth":"fHVoYy1wb29sLTgxNGFiODI0LTgwMzctNG
JjMy1iMTA2LWUwZDQyMWY5NDU2ZjpleUpoYkdjaU9pSlNVelV4TWlKOS5leUp6ZFdJaU9pSTJNekZpWW1Ka
09XTXlOVGswWTJaak9UWm1abUkwWVRRMU4yTmhORE00TVNKOS5jdVU0TjA4R3Q5VUh6Snk0d3VXOV9aeFdT
c1FiN0dUYThWRm0yTTdqWFBmSXd3TXZ0YnZGTHVwYno5bTZZX3VXTE91UF9SZk9xSVpldGk1X3oxUnBNOFV
nRDMzUUJ2V2QxbGFxMkJEY0ZUTHlyQ2Z5anVvaGdvZFppZE5SRFgwQXRURFdDLTlOMmRlMTZMUWhaeUttYn
BaZlFDVVo5TGtDLWdQSk1xVmNTOFY3RzhXM2x0QWkyTm55TGp1dzBDMk5MUjl3M3R4M29tRnhES1Jabm1kY
mFQemdzQ2o4bnNCZXhDYmtYRFVCVWo1YUFjX1Fha3FxNVZZYmtxYTQtNHRZYTljamc3RXEtUy11dW5pWlo3
cldxLUhOdmlYRUpWVVY0M1NkdnlNZ2I0bTQwanFaYVpqOVpLQ1pycktVUdnd0tTS1IteUNwcHhWZWZScnB
0QmZ4WndrM0ZPNGg1azZuOUdOSE1RazdmRGVnTEJ5WVdmcVdQNnZHSENudW1LRVFBLWczcXBPZENrWlptV2
VRQlpHaExHNkhpS0luRHVIOUxKb2hhSm5BcTBqVVhsVHdGYWZ4OUtoVXhUemM1cDhBUmQwZGdEd2tUZjJaV
WZKM2czVW1QUjNHaTRmY2xTaHc5RW9UbXlKXzYzdHpSV0pJd053X0VSOE9QSHVnNFNwM2pfZ2lPaXNCSmI2
TU5ybkJ3WmFXWElDNjdCbnR2REJVTXgyZkdYbDJ2cDZEamdpWjZxU2Z5blRwWXFpaFBGSWplbldQd0VqUFl
5b01nQTF2NVJoanppT1ZCemZXRTVKY1pOdC1VaU5GckdaMmV0bXdQUkpNaThJODJ2YWx3SjlxYkNZT21fQ2
xsRWI4TEF0aWJDTGtyYUVPRmY2cmpyMGRSUXZ1VlRIb3BJTQ==","email":"daniele.bagiotti@credi
t-agricole.it"}}}'
sshKey: 'ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIAG9AnCEFAut5NP+i5mHwh08c/0qTwLWRTZXIZwhhW+Z
root@GRPI-OCP-PV00'
```


L'*install-config.yaml* è stato posizionato sia all'interno della directory **/root/ocp-acmrz** che **/root/ocp-acmrz/install_dir.** L'installazione è stata lanciata con il comando seguente:

Python

```
[root@GRPI-OCP-PV00 ocp-acmrz]# cp install-config.yaml ./install_dir/.

[root@GRPI-OCP-PV00 ocp-acmrz]# ./openshift-install create cluster --dir
/root/ocp-acmrz/install_dir --log-level  debug
....
INFO Checking to see if there is a route at openshift-console/console...
DEBUG Route found in openshift-console namespace: console
DEBUG OpenShift console route is admitted
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/root/ocp-acmrz/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here:
https://console-openshift-console.apps.ocp-acmrz.cariprpc.it
INFO Login to the console with user: "kubeadmin", and password: "XXXXXXXXXXX"
```

```
DEBUG Time elapsed per stage:
DEBUG      pre-bootstrap: 38s
DEBUG          bootstrap: 12s
DEBUG             master: 17s
DEBUG Bootstrap Complete: 24m12s
DEBUG                API: 5m30s
DEBUG  Bootstrap Destroy: 1m14s
DEBUG  Cluster Operators: 18m22s
INFO Time elapsed: 45m2s
[root@grpi-ocp-hv00 ocp-acm]#
```

Al termine dell'installazione è possibile settare la variabile di ambiente KUBECONFIG per autenticarsi sul cluster appena creato:

Python
```
[root@GRPI-OCP-PV00 ocp-acmrz]# export
KUBECONFIG=/root/ocp-acmrz/install_dir/auth/kubeconfig
```

Python
```
[root@GRPI-OCP-PV00 ocp-acmrz]# oc whoami
```

ATTENZIONE: Dato che sarà utilizzato lo stesso Bastion anche per la successiva installazione del cluster di Rozzano abbiamo preferito creare un alias contenente il suddetto comando di export

Modificare il seguente file:

Python
```
[root@GRPI-OCP-PV00 ocp-acmac]# vi ~/.bashrc
```

Aggiungere le seguenti entry:

```
Python
alias ocp-acmac=export KUBECONFIG=/root/ocp-acmac/install_dir/auth/kubeconfig
alias ocp-acmrz=export KUBECONFIG=/root/ocp-acmrz/install_dir/auth/kubeconfig
```

Ricaricare la sessione Bash

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# source ~/.bashrc
```

Collegarsi al cluster

```
Python
[root@GRPI-OCP-PV00]# ocp-acmrz
```

Al termine della fase di installazione avremo un cluster composto da 3 nodi master e da 3 nodi worker.
In una fase successiva andremo a sostituire i nodi worker con dei nodi infrastrutturali.

La corretta installazione del cluster può essere verificata con il seguente comando

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# oc get clusteroperators
```

L'output conterrà lo stato del cluster e la colonna "AVAILABLE" dovrà essere "True"

2.5. Installazione ACM Rozzano

Al termine dell'installazione minimale è stato installato l'operator "*Advanced Cluster Management for Kubernetes*" utilizzando la dashboard.

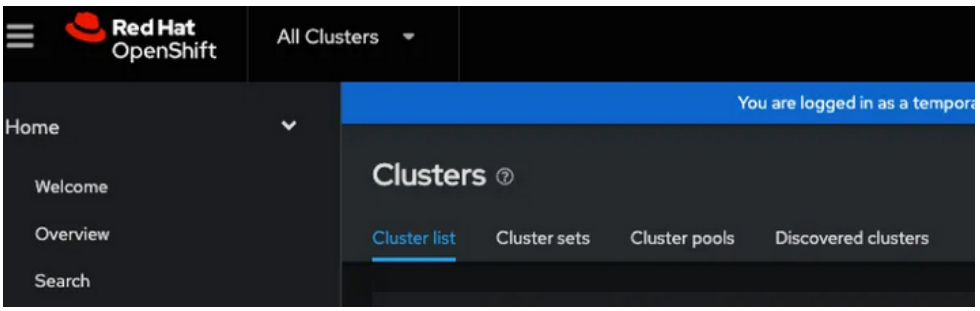
I passi eseguiti sono i seguenti:

- Collegandosi alla console di OCP del cluster ocp-acmrz, all'interno del tab "Operator Hub", digitare "Advanced Cluster Management for Kubernetes" e cliccare su "install".
- Nella pagina della subscription che si apre, lasciare il default "open-cluster-management" come namespace di default per l'installazione.
- Modalità di aggiornamento: selezionare "Automatic".

Una volta installato l'operator, selezionarlo e modificare la strategia di aggiornamento della subscription, da "Automatic" a "Manual".

Cliccando all'interno della sezione "MultiClusterHub" creare l'oggetto "MultiClusterHub" lasciando i valori di default.

Terminata l'installazione, è possibile collegarsi alla console di ACM utilizzando il tab "All Clusters" che si trova in alto a sinistra della console di OCP.



2.6. Configurazioni aggiuntive

Le cosiddette “operazioni di Day2”, sono state eseguite sul cluster OCP-ACMRZ utilizzando le policy di ACM.

Per semplificare la definizione e la gestione delle policy, è stato utilizzato il PolicyGenerator.

Seguono i comandi eseguiti per configurare il plugin del PolicyGenerator sul nodo bastion:

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# mkdir -p
${HOME}/.config/kustomize/plugin/policy.open-cluster-management.io/v1/policygenerat
or
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz ~]# wget
https://github.com/open-cluster-management-io/policy-generator-plugin/releases/down
load/v1.13.0/linux-amd64-PolicyGenerator
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# chmod +x linux-amd64-PolicyGenerator
```

```
Python
[root@GRPI-OCP-PV00 ocp-acmrz]# mv linux-amd64-PolicyGenerator
${HOME}/.config/kustomize/plugin/policy.open-cluster-management.io/v1/policygenerat
or/PolicyGenerator
```

All’interno della directory “**/root/cluster-acm-policy-generator/acm-hub/ocp-acmrz**” sono state create le policy specifiche per il cluster ocp-acmrz, nei prossimi paragrafi verranno descritte puntualmente.

Inoltre all’interno della directory “**/root/cluster-acm-policy-generator/acm-hub/all-cluster**” sono state create le policy valide sia per il cluster **ocp-acmac** che per il cluster **ocp-acmrz** quali NTP , Ldap sync, Oauth e ODF Operators.

2.6.1. Servizio Chronyd

All'interno della directory "**all-cluster/ntp"** sono stati definiti i template necessari a configurare il servizio *chronyd* sulle macchine virtuali di OCP.

Posizionarsi nella cartella contenente la policy "**/root/cluster-acm-policy-generator/acm-hub/all-cluster/ntp"**

```
Python
[root@GRPI-OCP-PV00 ntp]# cd
/root/cluster-acm-policy-generator/acm-hub/all-cluster/ntp
```

```
Python
[root@GRPI-OCP-PV00 ntp]# ll

-rw-r--r--. kustomization.yaml
-rw-r--r--. ntp-conf.yaml
-rw-r--r--. ntp-master-conf.yaml
-rw-r--r--. ntp-worker-conf.yaml
```

```
Python
[root@GRPI-OCP-PV00 ntp]# cat kustomization.yaml

generators:
  - ntp-conf.yaml
```

```
Python
[root@GRPI-OCP-PV00 ntp]# cat ntp-conf.yaml

apiVersion: policy.open-cluster-management.io/v1
kind: PolicyGenerator
metadata:
  name: generator-ntp-conf-ocp
placementBindingDefaults:
  name: placement-binding-ntp-conf-ocp
policyDefaults:
```

```
  namespace: acm-hub-policy
  placement:
    clusterSelectors:
      name: local-cluster
  complianceType: musthave
  remediationAction: enforce
  severity: high
policies:
  - name: policy-ntp-master-ocp
    manifests:
      - path: ntp-master-conf.yaml
  - name: policy-ntp-worker-ocp
    manifests:
      - path: ntp-worker-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 ntp]# cat ntp-master-conf.yaml

# Generated by Butane; do not edit
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: master
  name: 99-master-custom-ntp
spec:
  config:
    ignition:
      version: 3.4.0
    storage:
      files:
        - contents:
            compression: gzip
            source:
data:;base64,H4sIAAAAAAAAC/3zLMQ7CMAwAwN2v8AsSShlgRGJl4gWpmxaLUEeOQervkVDFhDyfrooUvN
7Oly5QUq5aKbAhDy9tBj/du9q7enD16Orpn47Kk01cMsZ30lh4iHRXWdb4FXimR26WK3Zhhz2oUVsXgiLzy
LodmbcDnwAAAP//yNra1AIBAAA=
          mode: 420
          overwrite: true
          path: /etc/chrony.conf
```

Python

```
[root@GRPI-OCP-PV00 ntp]# cat ntp-worker-conf.yaml

# Generated by Butane; do not edit
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
```

```
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-custom-ntp
spec:
  config:
    ignition:
      version: 3.4.0
    storage:
      files:
        - contents:
            compression: gzip
            source:
data:;base64,H4sIAAAAAAAC/3zLMQ7CMAwAwN2v8AsSShlgRGJl4gWpmxaLUEeOQervkVDFhDyfrooUvN
7Oly5QUq5aKbAhDy9tBj/du9q7enD16Orpn47Kk01cMsZ30lh4iHRXWdb4FXimR26WK3Zhhz2oUVsXgiLzy
LodmbcDnwAAAP//yNra1AIBAAA=
          mode: 420
          overwrite: true
          path: /etc/chrony.conf
```

N.B: Butane (precedentemente Fedora CoreOS Config Transpiler) è uno strumento che "legge" un file con sintassi Butane Config e produce una Ignition Config e verrà utilizzato per generare i MachineConfig di Openshift.

I file **ntp-master-conf.yaml** e **ntp-worker-conf.yaml** sono stati generati con l'utility "butane" come segue:

Scaricare il sorgente di Butane

```
Python
[root@GRPI-OCP-PV00 ntp]# curl
https://mirror.openshift.com/pub/openshift-v4/clients/butane/latest/butane --output
butane
```

Concedere i permessi di esecuzione

```
Python
[root@GRPI-OCP-PV00 ntp]# chmod +x butane
```

Spostarlo nella cartella /usr/local/sbin per renderlo utilizzabile sul bastion

```
Python
[root@GRPI-OCP-PV00 ntp]# mv butane /usr/local/sbin
```

Creare il file ntp-master.bu

```
Python
[root@GRPI-OCP-PV00 ntp]# cat ntp-master.bu

variant: openshift
version: 4.14.0
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 MSAD1.cariprpc.it iburst
          pool MSAD2.cariprpc.it iburst
          pool MSAD3.cariprpc.it iburst
          pool MSAD4.cariprpc.it iburst
          pool MSAD8.cariprpc.it iburst
          pool MSAD9.cariprpc.it iburst
          driftfile /var/lib/chrony/drift
          makestep 1.0 3
          rtcsync
          logdir /var/log/chrony
```

Creare il file ntp-worker.bu

```
Python
[root@GRPI-OCP-PV00 ntp]# cat ntp-worker.bu

variant: openshift
version: 4.14.0
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 MSAD1.cariprpc.it iburst
        pool MSAD2.cariprpc.it iburst
        pool MSAD3.cariprpc.it iburst
        pool MSAD4.cariprpc.it iburst
        pool MSAD8.cariprpc.it iburst
        pool MSAD9.cariprpc.it iburst
        driftfile /var/lib/chrony/drift
        makestep 1.0 3
        rtcsync
        logdir /var/log/chrony
```

Convertire il file ntp-master.bu in MachineConfig

```
Python

[root@GRPI-OCP-PV00 ntp]# /root/butane ntp-master.bu -o ./ntp-master-conf.yaml
```

Convertire il file ntp-worker.bu in MachineConfig

```
Python

[root@GRPI-OCP-PV00 ntp]# /root/butane ntp-worker.bu -o ./ntp-worker-conf.yaml
```

A questo punto sono state create le policy con il comando seguente:

```
Python
[root@GRPI-OCP-PV00 ntp]# oc kustomize --enable-alpha-plugins=true . | oc apply -f
-
```

IMPORTANTE: la configurazione del servizio *chronyd* prevede il riavvio di tutti i nodi del cluster in maniera *rolling*.
Attendere che tutti i nodi vengano riavviati prima di passare allo step successivo.

A seguito del riavvio è possibile verificare se il MachineConfig è stato applicato correttamente entrando in ssh/debug su un nodo e aprendo il file di configuzione di chronyd

```
Python
[root@GRPI-OCP-PV00 ntp]# ssh <ip node> -l core
```

```
Python
[root@GRPI-OCP-PV00 ntp]# cat /etc/chrony.conf
```

2.6.2. Definizione nodi infrastrutturali

All'interno della directory **/root/policy-generator/acm-hub/ocp-acmrz/infra-nodes** sono stati definiti i template necessari a:

- Eseguire il deploy dei nodi infrastrutturali
- Creare il *machineconfigpool* per i nodi infrastrutturali

```
Python
[root@GRPI-OCP-PV00]# cd
/root/cluster-acm-policy-generator/acm-hub/ocp-acmrz/infra-nodes
```

```
Python
[root@GRPI-OCP-PV00 infra-nodes]# ll
/root/cluster-acm-policy-generator/acm-hub/ocp-acmrz/infra-nodes

-rw-r--r--. acm-hub-infra-machineset.yaml
-rw-r--r--. infra-nodes-conf.yaml
-rw-r--r--. kustomization.yaml
-rw-r--r--. mcp-infra.yaml
```

Python

```
[root@GRPI-OCP-PV00 ntp]#  cat kustomization.yaml

generators:
  - infra-nodes-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 infra-nodes]#  cat infra-nodes-conf.yaml

apiVersion: policy.open-cluster-management.io/v1
kind: PolicyGenerator
metadata:
  name: generator-infra-node-conf-ocp
placementBindingDefaults:
  name: placement-binding-infra-node-conf-ocp
policyDefaults:
  namespace: acm-hub-policy
  placement:
    clusterSelectors:
      name: local-cluster
      datacenter: ROZZANO
  complianceType: musthave
  remediationAction: enforce
  severity: high
policies:
  - name: policy-infra-node-machineset-ocp
    manifests:
      - path: acm-hub-infra-machineset.yaml
  - name: policy-infra-node-mcp-ocp
    manifests:
      - path: mcp-infra.yaml
```

Python

```
[root@GRPI-OCP-PV00 infra-nodes]# cat acm-hub-infra-machineset.yaml

apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: ocp-acmrz-tsn8q
  name: ocp-acmrz-tsn8q-infra-0
  namespace: openshift-machine-api
spec:
  replicas: 3
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: ocp-acmrz-tsn8q
      machine.openshift.io/cluster-api-machineset: ocp-acmrz-tsn8q-infra-0
  template:
```

```
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: ocp-acmrz-tsn8q
        machine.openshift.io/cluster-api-machine-role: worker
        machine.openshift.io/cluster-api-machine-type: worker
        machine.openshift.io/cluster-api-machineset: ocp-acmrz-tsn8q-infra-0
    spec:
      lifecycleHooks: {}
      metadata:
        labels:
          node-role.kubernetes.io/infra: ""
      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_3_MGMT_AC
          numCPUs: 16
          numCoresPerSocket: 2
          snapshot: ""
          template: ocp-acmrz-tsn8q-rhcos-generated-region-generated-zone
          userDataSecret:
            name: worker-user-data
          workspace:
            datacenter: ROZZANO
            datastore: /ROZZANO/datastore/ESX-OCP-PROD-RZ-0000
            folder: /ROZZANO/vm/PRODUZIONE/OCP
            resourcePool: /ROZZANO/host/OCP_PROD//Resources
            server: rz-cags-vcsa001.cariprpc.it
```

Python

```
[root@GRPI-OCP-PV00 infra-nodes]# cat 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
```

Per creare le policy è stato eseguito il seguente comando:

```
Python
[root@GRPI-OCP-PV00 infra-nodes]# oc kustomize --enable-alpha-plugins=true . | oc
apply -f -
```

A questo punto è possibile verificare la creazione del *machineset* e del *machineconfigpool.*

```
Python
[root@GRPI-OCP-PV00 infra-nodes]# oc get machinesets -A

KKK

[root@GRPI-OCP-PV00 infra-nodes]# oc get mcp

KKK
```

IMPORTANTE: **Per aumentare la numerosità dei nodi infrastrutturali, andare in edit sul template della policy *acm-hub-infra-machineset.yaml* modificando il valore "replicas" e rilanciare il comando di create della policy, questo aggiornerà la policy con il nuovo valore.**

A questo punto è possibile eliminare i nodi "worker" dal cluster **ocp-acmrz** eseguendo i seguenti comandi:

Scalare a "0" il MachineSet dei nodi Worker

```
Python
[root@GRPI-OCP-PV00 infra-nodes]# oc scale --replicas=0 machineset
ocp-acmrz-tsn8q-worker-0
```

Eliminare definitivamente il MachineSet

```
Python

[root@GRPI-OCP-PV00 infra-nodes]# oc delete machineset ocp-acmrz-tsn8q-worker-0
```

N.B: I nodi verranno automaticamente prima Drenati e poi cancellati sia dal cluster Openshift sia da Vmware

2.6.3. Autenticazione tramite LDAP

All'interno della directory "**/root/cluster-acm-policy-generator/acm-hub/all-cluster/oauth"** sono stati definiti i template necessari a:

- Creare la *configmap* contenente la CA per l'utilizzo del protocollo ldaps nella comunicazione con il server LDAP
- Creare la secret contenente l'utenza per eseguire il bind all'LDAP e la password
- Configurare come metodo di autenticazione sul cluster, l'LDAP di Crédit Agricole

```
Python
[root@GRPI-OCP-PV00 oauth]# ll
/root/ocp-acm/cluster-acm-policy-generator/all-cluster/oauth

-rw-r--r--. auth-conf.yaml
-rw-r--r--. bind-secret.yaml
-rw-r--r--. kustomization.yaml
-rw-r--r--. cm-ca.yaml
-rw-r--r--. oauth.yaml
```

```
Python
[root@GRPI-OCP-PV00 oauth]# cat kustomization.yaml

generators:
  - auth-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 oauth ]#  cat auth-conf.yaml

apiVersion: policy.open-cluster-management.io/v1
kind: PolicyGenerator
metadata:
  name: generator-auth-conf-ocp
placementBindingDefaults:
  name: placement-binding-auth-conf-ocp
policyDefaults:
  namespace: acm-hub-policy
  placement:
    clusterSelectors:
      name: local-cluster
  complianceType: musthave
  remediationAction: enforce
  severity: high
policies:
  - name: policy-auth-conf-ocp
    manifests:
      - path: oauth.yaml
  - name: policy-auth-cm-ca-ocp
    manifests:
      - path: cm-ca.yaml
  - name: policy-auth-bind-secret-ocp
    manifests:
      - path: bind-secret.yaml
```

Python

```
[root@GRPI-OCP-PV00 oauth]# cat bind-secret.yaml

apiVersion: v1
data:
  bindPassword: <LDAP_BIND_PASSWORD>
kind: Secret
metadata:
  name: ldap-secret
  namespace: openshift-config
type: Opaque
```

Creare la ConfigMap contenente la Certificate Authority del server LDAP

Attenzione: Se dovesse cambiare va recreate questa ConfigMap

```
Python
[root@GRPI-OCP-PV00 oauth]# cat cm-ca.yaml

apiVersion: v1
data:
  ca.crt: |+
    <LDAP_CA >
kind: ConfigMap
metadata:
  name: ca-config-map
  namespace: openshift-config
```

```
Python
[root@GRPI-OCP-PV00 oauth]# cat oauth.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_4462_ocp_ldap,OU=OU-UTENTI-SERVIZI,OU=OU-UTENTI,DC=cariprpcpar,DC=it
      bindPassword:
        name: ldap-secret
      ca:
        name: ca-config-map
      insecure: false
      url:
ldaps://msad0par.cariprpcpar.it/DC=cariprpcpar,DC=it?sAMAccountName?sub?(&(objectCl
ass=user)(|(memberOf=CN=Users,DC=cariprpcpar,DC=it)(memberOf=CN=GU_DTR_USER,CN=User
s,DC=cariprpcpar,DC=it)(memberOf=CN=GU_OCP_ADMIN,CN=Users,DC=cariprpcpar,DC=it)(mem
berOf=CN=GU_OCP_USER,CN=Users,DC=cariprpcpar,DC=it)))
```

```
    mappingMethod: claim
    name: ldap
    type: LDAP
```

Per configurare l'operator dell'autenticazione è necessario eseguire il seguente comando:

```
Python
[root@GRPI-OCP-PV00 oauth]# oc kustomize --enable-alpha-plugins=true . | oc apply
-f -
```

Attendere il riavvio dei pod dell'autenticazione prima di testare la login tramite LDAP.

Verificare il riavvio corretto dei Pod di autenticazione:

```
Python
[root@GRPI-OCP-PV00 oauth]# oc get po -n openshift-authentication
```

2.6.4. Sync gruppi di utenti tra alberatura LDAP e OCP

All'interno della directory **/root/cluster-acm-policy-generator/acm-hub/all-cluster/group-sync-operator** sono stati definiti i template necessari a:

- Creare il namespace che ospita il group-sync-operator
- Installare il group-sync-operator
- Creare la *configmap* contenente la CA per l'utilizzo del protocollo ldaps nella comunicazione con il server LDAP: il campo ca.crt contiene il *base64encode* della CA.
- Creare la secret contenente l'utenza per eseguire il bind all'LDAP e la password
- Configurare il group sync operator
- Associare il ruolo cluster-admin agli utenti appartenenti al gruppo CN=GU_OCP_ADMIN,CN=Users,DC=cariprpc,DC=it

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# ll
/root/ocp-acm/cluster-acm-policy-generator/acm-hub/all-cluster/group-sync-operator

-rw-r--r--. group-sync-conf.yaml
-rw-r--r--. ldap-ca-bundle-group-sync.yaml
-rw-r--r--. ldap-creds-group-sync.yaml
-rw-r--r--. ldap-groupsync.yaml
-rw-r--r--. namespace.yaml
-rw-r--r--. operatorgroup.yaml
-rw-r--r--. role-bindig.yaml
-rw-r--r--. subscription.yaml
-rw-r--r--. kustomization.yaml
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat kustomization.yaml
generators:
  - group-sync-conf.yaml
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat group-sync-conf.yaml

apiVersion: policy.open-cluster-management.io/v1
kind: PolicyGenerator
metadata:
  name: generator-group-sync-conf-ocp
placementBindingDefaults:
  name: placement-binding-group-sync-conf-ocp
policyDefaults:
  namespace: acm-hub-policy
  placement:
    clusterSelectors:
      name: local-cluster
  complianceType: musthave
  remediationAction: enforce
  severity: high
policies:
  - name: policy-group-sync-namespace-ocp
    manifests:
      - path: namespace.yaml
```

```
  - name: policy-group-sync-operatorgroup-ocp
    manifests:
      - path: operatorgroup.yaml
  - name: policy-group-sync-subscription-ocp
    manifests:
      - path: subscription.yaml
  - name: policy-group-sync-ldap-groupsync-ocp
    manifests:
      - path: ldap-groupsync.yaml
  - name: policy-group-sync-ca-secret-ocp
    manifests:
      - path: ldap-ca-bundle-group-sync.yaml
  - name: policy-group-sync-bind-secret-ocp
    manifests:
      - path: ldap-creds-group-sync.yaml
  - name: policy-group-sync-admin-ocp
    manifests:
      - path: role-bindig.yaml
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat namespace.yaml

apiVersion: v1
kind: Namespace
metadata:
  name: group-sync-operator
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]#  cat operatorgroup.yaml

apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: group-sync-operator
  namespace: group-sync-operator
spec:
  targetNamespaces:
  - group-sync-operator
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat subscription.yaml

apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: group-sync-operator
  namespace: group-sync-operator
spec:
  channel: alpha
  installPlanApproval: Manual
  name: group-sync-operator
  source: community-operators
  sourceNamespace: openshift-marketplace
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]#  cat ldap-groupsync.yaml

apiVersion: redhatcop.redhat.io/v1alpha1
kind: GroupSync
metadata:
  name: ldap-groupsync
  namespace: group-sync-operator
spec:
  providers:
  - ldap:
      activeDirectory:
        groupMembershipAttributes:
        - memberOf
        userNameAttributes:
        - sAMAccountName
        usersQuery:
          baseDN: DC=cariprpc,DC=it
          derefAliases: never
          filter:
(&(objectClass=user)(|(memberOf=CN=Users,DC=cariprpc,DC=it)(memberOf=CN=GU_DTR_USER
,CN=Users,DC=cariprpc,DC=it)(memberOf=CN=GU_OCP_ADMIN,CN=Users,DC=cariprpc,DC=it)(m
emberOf=CN=GU_OCP_USER,CN=Users,DC=cariprpc,DC=it)))
          pageSize: 0
          scope: sub
      caSecret:
        kind: Secret
        name: ldap-ca-bundle-group-sync
        namespace: group-sync-operator
      credentialsSecret:
        kind: Secret
        name: ldap-creds-group-sync
        namespace: group-sync-operator
      insecure: false
```

```
      url: ldaps://msad1.cariprpc.it
      whitelist:
      - CN=Users,DC=cariprpc,DC=it
      - CN=GU_DTR_USER,CN=Users,DC=cariprpc,DC=it
      - CN=GU_OCP_ADMIN,CN=Users,DC=cariprpc,DC=it
      - CN=GU_OCP_USER,CN=Users,DC=cariprpc,DC=it
    name: ldap-group-sync
  schedule: '*/5 * * * *'
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]#  cat ldap-ca-bundle-group-sync.yaml

apiVersion: v1
data:
  ca.crt:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUZ0akNDQTU2Z0F3SUJBZ0lRTFl0ajZpdXQ0NTlPUnR
BV3JvYW40ekFOQmdrcWhraUc5dzBCQVFzRkFEQnMKTVFzd0NRWURWUVFHRXdKSlZERWZNQjBHQTFVRUNoTV
dRM0psWkdsMElFRm5jbWxqYjJ4bElFbDBZV3hwWVRFYQpNQmdHQTFVRUN4TVJVSEpwZG1GMFpTQkhjbTkxY
0NCUVMwa3hJREFlQmdOVkJBTVRGME55WldScGRFRm5jbWxqCmIyeGxTWFJoYkdsaFVrTkJNQjRYRFRFM01E
VXdNekUwTWpZeU9Wb1hEVFEzTURVd016RTBNell5Tmxvd2JERUwKTUFrR0ExVUVCaE1DU1ZReEh6QWRCZ05
WQkFvVEZrTnlaV1JwZENCQlozSnBZMjlzWlNCSmRHRnNhV0V4R2pBWQpCZ05WQkFzVEVWQnlhWFpoZEdVZ1
IzSnZkWEFnVUV0Sk1TQXdIZ1lEVlFREV4ZERjbVZrYVhSQlozSnBZMjlzClpVbDBZV3hwWVZKRFFUQ0NBa
Ul3RFFZSktvWklodmNOQVFFQkJRQURnZ0lQQURDQ0Fnb0NnZ0lCQU1wNzl4NnEKdmE2VHdaK0ZYNkRUdTIz
YzQyWjdWWmwrdDJxb01HN2NzY0J1VzNJVlRhbjBPdTV6d3hscGlXMnN5Tmx1OUJOMwo0Y2JHYm5CK2poeWc
vN0Z2NmRxbUZoRkpmQVBYM1JSMGpvZDUzS2ZqenZqZmMvSXBvZzExajhtZkFSMWwzTjY2CmFwU1krM0o5dU
5mTmpvT2xWZXUrU0hYVGxYbFVZNVJuNlp6a0E0VEM0TlJYeWY5b0Z3ZkJsem1tWXYvK1pHS0kKOWlUKzd6V
XZWM2Q5bHJCV1QybkN0N3A2eWhENG5ESUVzY0I4d0dvQ3ZScmtiTjJFdDJNVll2UFpYaHFhYVMxRwpLdi9M
S0ZlOU9LUjJyS1VHckxkbzB1aWcyQzVuQXBlTXF4K3lMNVhORGVUc1NwNEpjS0JNZmxMTVZ6UHFBSW4xClV
IUnc2cDVXdmtsb3ROaTFycDIvUkZucDFYenh3YkdkVTZwM2VXYVBtckMvZ0NnT1hqcW56Q1dwZDZiVW0xRT
YKclZFNkNON0tSU3FadWptTlZDb1BKQmtXaFVJeTdOSlJLT0N0bCtPVUxoNTVGV2VKbnJoS0ZjMWxOVVRWb
0tQNwpYVys2WE9pNGVyc1hHUUN5RTd3bE1ZenJMUDFSdFhrK2dWanZyUHhCMklsWHpMalZMRnZvN1FoL0RO
UHhaNEFJCng5WHVYa1IzMElqWnhkR2VlM0VVd2p0dmFoclg2U1BVV2w4NzIxY2ovTUgxTkd2Q2hHZTZ1c29
oa0ZCeXdKc24KS1R2a0U4U2prRDQ0R2dxZW1pRlNVMHBrN0Z0S2RSZ3RYYTFXa1VwWWwwMzhxUHpTaHdJbW
01SDRQQVpqU2k3cgowOHVCZ0JLdG9KcVJHL0Y1dzFoZkdwbDFxQnJKN3FPZXdQUVBBZ01CQUFHalZEQlNNQ
XNHQTFVZER3UUVBd0lCCmhqQVNCZ05WSFJNQkFmOEVDREFHQVFIL0FnRUJNQjBHQTFVZERnUVdCQlFYNlJl
anZDNVBLNTd2ejBxTmZCQ3YKbkd4KzB6QVFCZ2tyQmdFRUFZSTNGUUVFQXdJQkFEQU5CZ2txaGtpRzl3MEJ
BUXNGQUFPQ0FnRUFJeGhFbUUrdQpjYjZ0UzRPdENHUjhpb3VGK2xjYlFIRnpOUVlhUWxwRUE5c3FSekl2MW
Uvc2o5WnBFNDRsU1ZSakdMWVhnOHViCkVjVnVwWGxMeG5GVWt6dVJNVjhEc0RxZkR5T3NuM2hFS3dlOFJzc
nQzQUFTa1JycDFnOHZhcTlScGViTVM0WWgKNnRsa1BLVWJQQ0NDZXdjaER3dWIzQXpCU3EzRWkzOEk0Ui9I
VjRyRHpZK3N4SHRkRDV4UmNuaUdtWTJneTY5VwpLOHdoTUZEQ1RzdXJVYWdBNllluRURrQlRiRk0vTWlYUkh
UczBLUWE1R1I0cGVhY1FVREsrckJUWWloN3YvUG5yCnRMVFdhRnYyaTBJUVRGcWd4N0RJb1JWUzlIOFNLRE
pkamovbktSWmk0d3Y2b083TEUydG1qa2U3MzIzeElDejMKQjRaQmNKQVo2aUZ4NFhNTXJRNlowNE5IRUdHY
```

```
WtGQVV4Z2h3OGVJc3oyb2lKKzZRK2oxNmF4eEFldGprd1RMQgprY2lSanJ0N0FKYkFieEF2SWg5TFNOU201
K1NzQ2VhTDY0UjV6U2psMmV5NVBzT000dzdaN3J1dTU1M3BIcDEvCkkzdzEvZER2UTZteitRb0lHV0REbTd
zMDk1bHRSZEF6TlNuK00vaGVEZWxMSVdqZnhGMm4wNTJVQ0lOeERLL1MKZ1RLUmlnbDQ0NkxRL0cyc0tDWS
tYVFo0Vjk0aWp1Mk95R0pDdk5vTEtHMUdoS0hsWGJMQ3FpU3ZQRnE0bE9ZMwo1MTNBOXFEcnBXUjBEanBDb
TZoRjVaNXhSUHFvdnZhSmhtbStTd1oyMXpQMHY4M3lOS0tTanVub0ROUTZWakw3CkRSVWpMZGhVR2hhaW10
SGFnSnFkWlV4dlRsNUhBQit0N1F3PQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCi0tLS0tQkVHSU4gQ0V
SVElGSUNBVEUtLS0tLQpNSUlJVWpDQ0JqcWdBd0lCQWdJVEtnQUFBQUpoelBJNHA0T2YrQUFBQUFBQUFqQU
5CZ2txaGtpRzl3MEJBUXNGCkFEQnNNUXN3Q1FZRFZRUUdFd0pKVkRFZk1CMEdBMVVFQ2hNV1EzSmxaR2wwS
UVGbmNtbGpiMnhsSUVsMFlXeHAKWVRFYU1CZ0dBMVVFQ3hNUlVISnBkbUYwWlNCSGNtOTFjQ0JRUzBreElE
QWVCZ05WQkFNVEYwTnlaV1JwZEVGbgpjbWxqYjJ4bFNYUmhiR2xoVWtOQk1CNFhEVEUzTURVd016RTFNRGN
5TjFvWERUSTVNRFV3TXpFMU1UY3lOMW93CmJERUxNQWtHQTFVRUJoTUNTVlF4SHpBZEJnTlZCQW9URmtOeV
pXUnBkQ0JCWjNKcFkyOXNaU0JKZEdGc2FXRXgKR2pBWUJnTlZCQXNURVZCeWFYWmhkR1VnUjNKdmRYQWdVR
XRKTVNBd0hnWURWUVFERXhkRGNtVmthWFJCWjNKcApZMjlzWlVsMFlXeHBZVk5EUVRDQ0FpSXdEUVlKS29a
SWh2Y05BUUVCQlFBRGdnSVBBRENDQWdvQ2dnSUJBTEpZCjM3dG9BZFBYM2lTVnZxTHA3RHFqQjBEWXllVWp
BcVBnVnh0V2JGbU9CelBpMHo4S3Q1bUs4MmxIUi9Db3FxTlAKUU9MWjZIbUY4dTlqU3ljNGlaRjZkU2V2OF
YzQkxDVWEzWmZmYjlkM2hIb2cwMWdhcnFIbUlpaytOVjlRNTl1WApIZjFCOTRmaUJ1UmFhdVVKTTVHUG05O
Dl5dnBqT0NLa2ppaUxlQ2dBSG5EQUtnZ3c3N2twcEVUYzMxUklNWklkCmR1TzZ4K2FoMmZHeUE2NEZOVkJp
UDdVa0RXK2djNkhiUEx4UDA2QTRuZmc1MnI5a2NSL0x5UUxQVXM3WmxYLzcKNjR2d2MwNUtwZ1RHU3hYTml
JdFZjUnRMSm9wWFlBYlRkZ0ZHUFl6L05JQkorZTRwMHNoY29VRnNDREhUSGZpegozcEFXODQxb3pyZWpRQn
ZQekYxcXJCZEQ2OFFXKzBGODUrZXV1QStITUtEYXR1a0g4Y3lURWJUMzVpalFmTGdlCkE0YjA3TGh3R2ljd
GN2RU1qWE15c25kV29yK2JLS2NTMG9FSVBWbGpad0t4bHUzTGZkMHRVOTZIYVVDQktYWmcKUkVwd1A2SWdQ
aDhhVFFLOXBCN0h2cW1CTmp5elkvMU9UVkFJYld1N1N6WWN0RWx2cmFCYjRIRzVVRzEvVEhyNgprMjVzUk8
yNlpjVEp6T3RXNWVlRnM1T0haMER2Qk5oYk1aYnJEMjlJWUxOaTdNdU5YcWpZOEpWRHFnSEtEeGVxCnV3M1
dmZHdFS2NGd1YySnVXNVN6RVZVREhzY3kzQTZXcm1wd3pFTlV6ejVvOXhKNklGc0lxQUE1TDkwdUlzanMKU
ytJOUFSc2ZqV29GdDNJNXUxbGlCMUtPaXZhREVFbnFXbnJKYXdoMUFnTUJBQUdqZ2dMck1JSUM1ekFRQmdr
cgpCZ0VFQVlJM0ZRRUVBd0lCQURBZEJnTlZIUTRFRmdRVTRBclpuZEpodjZQeVI5Z3RFcU5kNFlyNHRIc3d
HUVlKCkt3WUJCQUdDTnhRQ0JBd2VDZ0JUQUhVQVlnQkRBRUV3Q3dZRFZSMFBCQVFEQWdHR01CSUdBMVVkRX
dFQi93UUkKTUFZQkFmOENBUUF3SHdZRFZSMGpCQmd3Rm9BVUYra1hvN3d1VHl1ZTc4OUtqWHdRcjV4c2Z0T
XdnZ0VtQmdOVgpIUjhFZ2dFZE1JSUJHVENDQVJXZ2dnRVJvSUlCRFlhQnhHeGtZWEE2THk4dlEwNDlRM0ps
WkdsMFFXZHlhV052CmJHVkpkR0ZzYVdGU1EwRXNRMDQ5UjFKUVNTMURRVUV0VUZZek1TeERUajFEUkZBc1E
wNDlVSFZpYkdsakpUSXcKUzJWNUpUSXdVMlZ5ZG1salpYTXNRMDQ5VTJWeWRtbGpaWE1zUTA0OVEyOXVabW
xuZFhKaGRHbHZiaXhFUXoxRApRVkpKVUZKUVF5eEVRejFwZEQ5alpYSjBhV1pwWTJGMFpWSmxkbTlqWVhSc
GIyNU1hWE4wUDJKaGMyVS9iMkpxClpXTjBRMnhoYzNNOVkxSk1SR2x6ZEhKcFluVjBhVzl1VUc5cGJuU0dS
R2gwZEhBNkx5OXdhMmt1WTNKbFpHbDAKTFdGbmNtbGpiMnhsTG1sMEwwTmxjblJGYm5KdmJHd3ZRM0psWkd
sMFFXZHlhV052YkdWSmRHRnNhV0ZTUTBFdQpZM0pzTUlJQkt3WUlLd1lCQlFVSEFRRUVnZ0VkTUlJQkdUQ0
J0Z1lJS3dZQkJRVUhNQUtHZ2Fsc1pHRndPaTh2CkwwTk9QVU55WldScGRFRm5jbWxqYjJ4bFNYUmhiR2xoV
WtOQkxFTk9QVUZKUVN4RFRqMVFkV0pzYVdNbE1qQkwKWlhrbE1qQlRaWEoyYVdObGN5eERUajFUWlhKMmFX
TmxjeXhEVGoxRGIyNW1hV2QxY21GMGFXOXVMRVJEUFVOQgpVa2xRVWxCRExFUkRQV2wwUDJOQlEyVnlkR2x
tYVdOaGRHVS9ZbUZ6WlQ5dllltcGxZM1JEYkdGemN6MWpaWEowCmFXWnBZMkYwYVc5dVFYVjBhRzl5YVhSNU
1GNEdDQ3NHQVFVRkJ6QUNobEpvZEhSd09pOHZjR3RwTG1OeVpXUnAKZEMxaFozSnBZMjlzWlM1cGRDOURaW
EowUlc1eWIyeHNMMGRTVUVrdFEwRkJMVkJXTXpGZlEzSmxaR2wwUVdkeQphV052YkdWSmRHRnNhV0ZTUTBF
dVkzSjBNQTBHQ1NxR1NJYjNEUUVCQ3dVQUE0SUNBUUM5UTd3cGFwRnQ0Wk9XCmJaMnZaN0pjUGMwYUtRZnl
oMVd1clBYTk1JaGdxTkdackVXdHYrVGRtQ3Z5VHk3eW03Z0JrM1ZyQVhtWVhYdDUKZ3FlRnl0dHZCdEFMOD
Nnd05BWmNoSmxqWVFsYXJPek1WSXNnTVluUGFjK29xeTI1dVI1MHlCenI1ek8rZ1BjYgo2TzNLQzRLL200e
TdRMFhzSDNNa2VMMXpmRkgwYkFNMnFMYnVvcE5kY3hTWGw2bVFES2dENUJBV0xGUDFGbGtYCkxZLzdmQzh6
R0o4SjlnZVJrOFZGR3VQL1N2RnZvcW5DUHBFYU42RmVLbWtBaWxZTGlwZjNkUjU5YUJ5SUNYWksKWGhLTld
VWG81dTZxdW1ZZlNDbW5SbWVCZXRDTnNiRGt4aVd3UXZOd1pXY1diMG45elFNUzJCRnRTS0loMGpJVApzWD
dkTUtQRnFPQ0I3SHcyRi8zdmh5b3JDeFA1K0d3eDVDS0UwRWdiRkVoVUd3YmNrTHQ3OW05dDVtOWF3bGJhC
mdiVll4ampyNzhpd21icXQwejlXd3RERHhZVldpVDFSSmtmV3E4R3JNLzlIaHVOS2ZUbVZYUVFyaW1uRXNh
T2gKNllVU281N2prZSt1NHFQL25CalowdVVOaGV6WmV4N3hDRGF0aDRKckVEZEZaNFJ2MjRKNWhqWTlFaUF
mbHdQSApwM3V5TjhLNHRMV3VKdmU0N3dhZnQzaEU2WUJHK1V2NTdxbUdRbllESGt1NzVRbmErd1ljOFRJM1
R2c3ZKZWlyCk9PSjlpeHpaYUhoMEZFbkpXVmU0Y0xPM2preEZlb09EM3B1WVE4VXJPVUJwZDVQY3phNVg2a
mJVNE4zdjNXOFgKcHFsQUx0Tmg4WnR6VXlibVVpK1RBbmpvY092akV3PT0KLS0tLS1FTkQgQ0VSVElGSUNB
VEUtLS0tLQo=
kind: Secret
metadata:
  name: ldap-ca-bundle-group-sync
```

```
  namespace: group-sync-operator
type: Opaque
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]# cat ldap-creds-group-sync.yaml

apiVersion: v1
data:
  password: <LDAP_BIND_PASSWORD>
  username: <LDAP_BIND_USER>
kind: Secret
metadata:
  name: ldap-creds-group-sync
  namespace: group-sync-operator
type: Opaque
```

Python

```
[root@GRPI-OCP-PV00 group-sync-operator]#  cat role-bindig.yaml

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: clusteradmin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: CN=GU_OCP_ADMIN,CN=Users,DC=cariprpc,DC=it
```

Per creare questa policy è stato lanciato il comando seguente:

Python

```
[root@grpi-ocp-hv00 group-sync-operator]# oc kustomize --enable-alpha-plugins=true
. | oc apply -f -
```

IMPORTANTE: una volta creata la policy, collegarsi alla console di OCP, ed approvare a mano l'installation-plan dell'operator group-sync-operator.

2.6.5. Installazione ODF

I passi eseguiti sono i seguenti:

- Collegandosi alla console di OCP del cluster ocp-acmrz, all'interno del tab "Operator Hub", digitare "OpenShift Data Foundation" e cliccare su "install".
- Nella pagina della subscription che si apre, lasciare il default "openshift-storage" come namespace di default per l'installazione.
- Modalità di aggiornamento: selezionare "Automatic".
- Abilitare il plugin della console
- Cliccare su Install

Una volta installato l'operator, selezionarlo e modificare la strategia di aggiornamento della subscription, da "Automatic" a "Manual".

2.6.6. Configurazione ODF

I passi eseguiti sono i seguenti:

- Aprire operator OpenShift Data Foundation
- Creare StorageSystem
- Selezionare "Full Deployment"
- Selezionare la StorageClass di Vmware "thin-csi"
- Lasciare il resto di Default
- Attendere completamento dell'installazione

Una volta completata l'installazione creare un nuovo pvc per testare le nuove StorageClass.

2.6.7. Aumento replica IngressController a 3

Dato che i nodi Infra previsti sono con replica 3 è buona norma incrementare anche le repliche del pod Ingress

```
Python
[root@GRPI-OCP-PV00 group-sync-operator]#
```

```
oc patch ingresscontroller/default -n openshift-ingress-operator --type=merge -p
'{"spec":{"replicas": 3}}'
```

3. Configurazione Active Passive

La configurazione Active Passive effettuata prevede che il cluster Active sia quello di Acilia mentre il Passive quello di Rozzano. La configurazione può essere invertita agendo opportunamente sui cluster.

3.1. Policy

L'istanza ACM del cluster Acilia è stata utilizzata per definire e applicare un intero set di policy sia ai cluster gestiti che al cluster locale, identificato con il selettore del cluster 'local-cluster'. Le policy sono state create utilizzando il **PolicyGenerator** (vedi Configurazioni aggiuntive)

Per eseguire il backup delle policy ACM è necessario seguire una serie di precauzioni:

- Se ci sono policy che utilizzano '**local-cluster**' come **clusterSelector**, queste devono essere escluse dal backup; in caso contrario, verrebbero applicate al '**local-cluster**' passivo durante la fase di ripristino, a meno che non si tratti di una scelta intenzionale.
- Per escludere una policy dal backup, è necessario aggiungere la seguente label alla policy in questione:

```Python
policyLabels:
  velero.io/exclude-from-backup: "true"
```

- Considerando che anche il cluster ACM passivo avrà policy applicate, per evitare confusione tra il 'local-cluster' attivo e quello passivo, è preferibile utilizzare nella definizione delle policy un **ClusterSelector** che selezioni un cluster in base a un'etichetta creata specificamente, ad esempio *ClusterName*.

3.2. Configurazione CA-Bundle

L'architettura Active-Passive utilizzando lo storage S3 fornito da ODF prevede che sia affidabile la comunicazione tra gli endpoint S3 tra i due cluster di Acilia e Rozzano.

Risulta necessario quindi aggiungere nei truststore di ciascun cluster OCP la CA del complementare.

Per recuperare la CA attualmente configurato dai router OCP seguire i seguenti passi:

```
Python
--- Creazione directory di lavoro Rozzano ---

[root@GRPI-OCP-PV00 ~]# mkdir cluster-acm-dr && cd cluster-acm-dr
[root@GRPI-OCP-PV00 cluster-acm-dr]# mkdir acmrz-passive && cd acmrz-passive

--- Accesso OCP Rozzano ---
[root@GRPI-OCP-PV00 acmrz-passive]# ocp-acmrz

--- Recupero CA di Acilia ---
[root@GRPI-OCP-PV00 acmrz-passive]# openssl s_client -showcerts -connect
console-openshift-console.apps.<acilia_clustername>.<acilia_domain>:443

> inserire il pem nel file ca-bundle-acmac.crt

[root@GRPI-OCP-PV00 acmrz-passive]# oc create configmap custom-ca
--from-file=ca-bundle.crt=./ca-bundle-acmac.crt -n openshift-config

[root@GRPI-OCP-PV00 acmrz-passive]# oc patch proxy/cluster --type=merge
--patch='{"spec":{"trustedCA":{"name":"custom-ca"}}}'



--- Creazione directory di lavoro Acilia ---

[root@GRPI-OCP-PV00 ~]# cd cluster-acm-dr
[root@GRPI-OCP-PV00 cluster-acm-dr]# mkdir acmac-active && cd acmac-active

--- Accesso OCP Acilia ---
[root@GRPI-OCP-PV00 acmac-active]# ocp-acmac

--- Recupero CA di Acilia ---
[root@GRPI-OCP-PV00 acmac-active]# openssl s_client -showcerts -connect
console-openshift-console.apps.<acilia_clustername> <rozzano_domain>:443

> inserire il pem nel file ca-bundle-acmrz.crt

[root@GRPI-OCP-PV00 acmac-active]# oc create configmap custom-ca
--from-file=ca-bundle.crt=./ca-bundle-acmrz.crt -n openshift-config
```

```
[root@GRPI-OCP-PV00 acmac-active]# oc patch proxy/cluster --type=merge
--patch='{"spec":{"trustedCA":{"name":"custom-ca"}}}'
```

3.3. Configurazione Bucket S3

Sull'HUB cluster di Rozzano, è stato installato MCG **(Multicloud Object Gateway)** per avere a disposizione uno storage S3 per memorizzare i backup di ACM.

Una volta installato l'operatore, è stato creato un bucket S3 utilizzando **ObjectBucketClaims**.

Sono stati utilizzati i seguenti manifest:

```
Python
--- Accesso OCP Rozzano ---

[root@GRPI-OCP-PV00 ~]# ocp-acmrz

--- Creazione S3 Bucket Rozzano ---

[root@GRPI-OCP-PV00 cluster-acm-s3]# cd acmrz-passive
[root@GRPI-OCP-PV00 acmrz-passive]# vim s3-obc-passive.yaml
---
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: acm-hub-backup
  namespace: openshift-storage
  labels:
    app: noobaa
spec:
  additionalConfig:
    bucketclass: noobaa-default-bucket-class
  generateBucketName: acm-hub-backup
  storageClassName: openshift-storage.noobaa.io

[root@GRPI-OCP-PV00 acmrz-passive]# oc apply -f s3-obc-passive.yaml
```

Una volta creato il bucket, è possibile estrarre le credenziali per il suo utilizzo.

- access-secret
- access-secret-key
- bucketname
- endpoint

Sono stati utilizzati i seguenti comandi per estrarre le informazioni, attenzione queste informazioni sono dinamiche e quindi cambiano alla creazione di ogni bucket:

```
Python
[root@GRPI-OCP-PV00 acmrz-passive]# oc get secret acm-hub-backup -n
openshift-storage -o jsonpath='{.data.AWS_ACCESS_KEY_ID}' | base64 --decode

> VWRhDb0yUpyPCif3Y3HC

[root@GRPI-OCP-PV00 acmrz-passive]# oc get secret acm-hub-backup -n
openshift-storage -o jsonpath='{.data.AWS_SECRET_ACCESS_KEY}' | base64 --decode

> NinP3SSOdBEt6nnyWgxcsgIJ12037jI9gCmPtl1x

[root@GRPI-OCP-PV00 acmrz-passive]# oc get obc acm-hub-backup -o yaml -n
openshift-storage | grep "bucketName" | awk '{print $2}'


> acm-hub-backup-5d0ad6e3-1c9d-4792-bd64-6207cd21c3a8

[root@GRPI-OCP-PV00 acmrz-passive]# oc get route -n openshift-storage | grep
"s3-openshift" | awk '{print $2}'


> https://s3-openshift-storage.apps.ocp-acmrz.cariprpc.it
```

Appuntare temporaneamente questi output per proseguire con l'attività.

3.4. Configurazione Acilia come active cluster

Al tempo "0", il cluster Acilia HUB gestisce una serie di cluster:

```
Python
--- Accesso OCP Acilia ---
[root@GRPI-OCP-PV00 ~]# ocp-acmac
[root@GRPI-OCP-PV00 ~]# oc get managedcluster

NAME                HUB ACCEPTED  MANAGED CLUSTER URLS
local-cluster       true          https://api.ocp-acmac.cariprpc.it:6443
ocp2-parallelo      true          https://api.ocp2-parallelo.cariprpcpar.it:6443
```

È inoltre possibile verificare dalla console di ACM che tutti i cluster siano nello stato "Ready".
Per configurare il backup sui cluster attivi, segui questi passaggi:

1. Dalla dashboard, seleziona l'operatore ACM.
2. Vai su "MultiClusterHub".
3. Modifica l'istanza esistente da Yaml, abilitando il backup del cluster.

```
Python
  components:
    - enabled: true
      name: cluster-backup
    ...
```

In alternativa da linea di comando eseguendo:

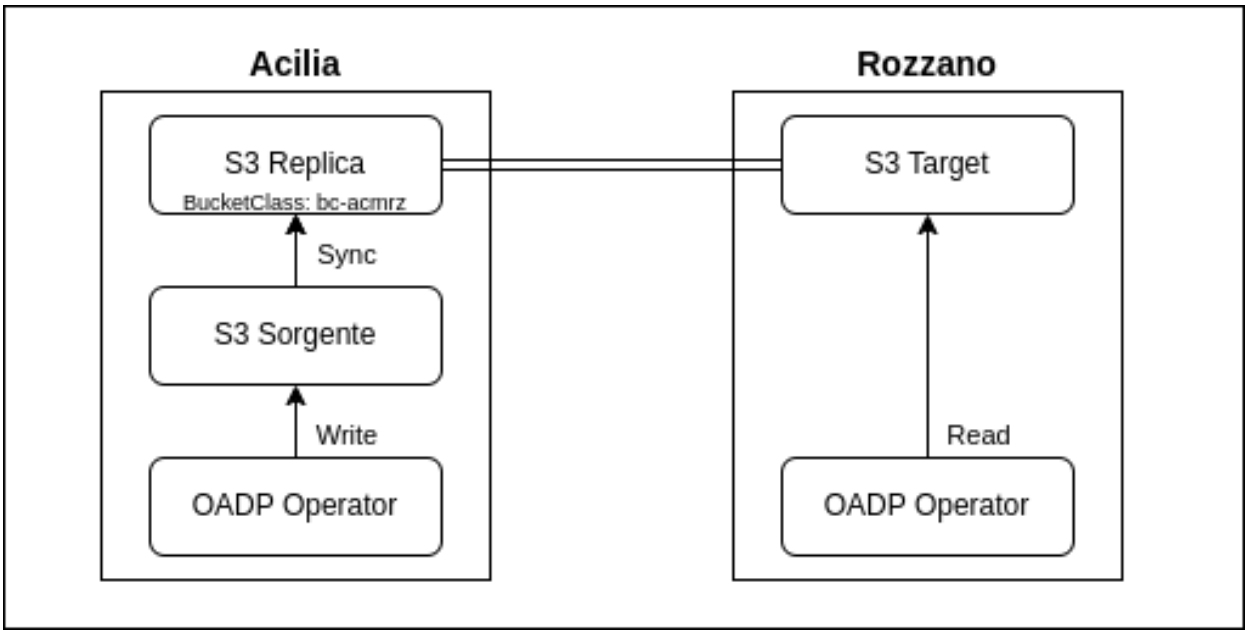
```
Python
[root@GRPI-OCP-PV00 ~]# oc edit multiclusterhub -n open-cluster-management

[...]
spec:
  availabilityConfig: High
  enableClusterBackup: false
  ingress:
    sslCiphers:
      - ECDHE-ECDSA-AES256-GCM-SHA384
      - ECDHE-RSA-AES256-GCM-SHA384
      - ECDHE-ECDSA-AES128-GCM-SHA256
      - ECDHE-RSA-AES128-GCM-SHA256
  overrides:
    components:
     [...]
      - enabled: true
        name: cluster-backup
     [...]
```

Dopo alcuni minuti, è possibile verificare nell'elenco degli operatori installati che:

- L'operator **OADP** esiste.
- È installato all'interno del namespace: **open-cluster-management-backup**.

```
Python
[root@GRPI-OCP-PV00]# oc get subscriptions.operators.coreos.com  -n
open-cluster-management

NAME                          PACKAGE                       SOURCE
CHANNEL
advanced-cluster-management   advanced-cluster-management   redhat-operators
release-2.10
```

Passare alla creazione della replica S3 per consentire la sincronizzazione dei backup di ACM da Acilia verso Rozzano:



1. Predisposizione del **NamespaceStore:**

La CRD **NamespaceStore** rappresenta un target storage da utilizzare come storage sottostante per i dati nei bucket di NooBaa. I **NamespaceStores** sono successivamente referenziati per nome da una **BucketClass**. I **NamespaceStore** necessitano delle credenziali **AWS_ACCESS_KEY_ID** e **AWS_SECRET_ACCESS_KEY** dello storage target:

```
Python
--- Creazione secret credenziali per accesso a bucket S3 Rozzano ---

[root@GRPI-OCP-PV00 cluster-acm-dr]# cd acmac-active
[root@GRPI-OCP-PV00 acmac-active]# vim s3-credential_ns-passive.yaml

apiVersion: v1
stringData:
  AWS_ACCESS_KEY_ID: VWRhDb0yUpyPCif3Y3HC
  AWS_SECRET_ACCESS_KEY: NinP3SSOdBEt6nnyWgxcsgIJ12037jI9gCmPtl1x
kind: Secret
metadata:
  name: ns-credential-passive
  namespace: openshift-storage
type: Opaque

[root@GRPI-OCP-PV00 ocp-acmac]# oc apply -f s3-credential_ns-passive.yaml


--- Creazione oggetto NamespaceStore ---

[root@GRPI-OCP-PV00 acmac-active]# vim s3-ns-passive.yaml

apiVersion: noobaa.io/v1alpha1
kind: NamespaceStore
```

```
metadata:
  name: ns-passive
  namespace: openshift-storage
  labels:
    app: noobaa
spec:
  s3Compatible:
    endpoint: 'https://s3-openshift-storage.apps.ocp-acmrz.cariprpc.it'
    secret:
      name: ns-credential-passive
      namespace: openshift-storage
    targetBucket: acm-hub-backup-5d0ad6e3-1c9d-4792-bd64-6207cd21c3a8
  type: s3-compatible

[root@GRPI-OCP-PV00 acmac-active]# oc apply -f s3-ns-passive.yaml
[root@GRPI-OCP-PV00 acmac-active]# oc get namespacestore ns-passive -n
openshift-storage

> N.B Verificare che sia in status "Available"
```

2. Definizione della **BucketClass:**

La CRD **BucketClass** rappresenta una struttura che definisce le politiche del bucket relative al posizionamento dei dati, come il riferimento ad un **NamespaceStore**.

```
Python
--- Creazione oggetto BucketClass ---

[root@GRPI-OCP-PV00 acmac-active]# vim s3-bz-passive.yaml

apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  name: bc-passive
  namespace: openshift-storage
  labels:
    app: noobaa
spec:
  namespacePolicy:
    single:
      resource: ns-passive
    type: Single

[root@GRPI-OCP-PV00 acmac-active]# oc apply -f s3-bc-passive.yaml
[root@GRPI-OCP-PV00 acmac-active]# oc get bucketclass bc-passive -n
openshift-storage

> N.B Verificare che sia in status "Available"
```

3. Definizione **ObjectBucketClaim** su cluster ACM Acilia con riferimenti alla **BucketClass** creata in precedenza:

```
Python
--- Creazione oggetto ObjectBucketClaim su BucketClass riferita a Rozzano ---

[root@GRPI-OCP-PV00 acmac-active]# vim s3-obc-passive-rep.yaml

apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: acm-hub-backup-passive
  namespace: openshift-storage
  labels:
    app: noobaa
spec:
  additionalConfig:
    bucketclass: bc-passive
  generateBucketName: acm-hub-backup-passive
  storageClassName: openshift-storage.noobaa.io

[root@GRPI-OCP-PV00 acmac-active]# oc apply -f s3-obc-passive-rep.yaml
[root@GRPI-OCP-PV00 acmac-active]# oc get obc acm-hub-backup-passive -n
openshift-storage
> N.B Verificare che sia in status "Bound"

--- Recuperare il BucketName per azione successiva ---

[root@GRPI-OCP-PV00 acmac-active]# oc get obc acm-hub-backup-passive -n
openshift-storage -o yaml | grep "bucketName" | awk '{print $2}'

> acm-hub-backup-passive-cdd9116d-0dbf-4f9c-9c75-d90edeea25e2
```

4. Creare **ObjectBucketClaim** utilizzato da OADP Operator per storicizzare i backup prodotti da ACM di Acilia:

Questo bucket possiamo denominarlo come “Bucket Sorgente” perchè viene configurato con le istruzioni per replicare i dati verso il bucket che utilizza la **BucketClass** con puntamento a Rozzano:

```
Python
--- Creazione oggetto ObjectBucketClaim sorgente ---

[root@GRPI-OCP-PV00 acmac-active]# vim s3-obc-active.yaml

apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: acm-hub-backup
```

```
  namespace: openshift-storage
  labels:
    app: noobaa
spec:
  additionalConfig:
    bucketclass: noobaa-default-bucket-class
    replicationPolicy: |
      {
        "rules": [
          {"rule_id": "rule-1", "sync_deletions": true, "destination_bucket":
"acm-hub-backup-passive-cdd9116d-0dbf-4f9c-9c75-d90edeea25e2"}
        ]
      }
  generateBucketName: acm-hub-backup
  storageClassName: openshift-storage.noobaa.io

[root@GRPI-OCP-PV00 acmac-active]# oc apply -f s3-obc-active.yaml
[root@GRPI-OCP-PV00 acmac-active]# oc get obc acm-hub-backup -n openshift-storage

> N.B Verificare che sia in status "Bound"

--- Recuperare il BucketName e credenziali per step 5 e 6 ---

[root@GRPI-OCP-PV00 acmac-active]# oc get secret acm-hub-backup -n
openshift-storage -o jsonpath='{.data.AWS_ACCESS_KEY_ID}' | base64 --decode

> 5AHJv5hLEmOd7wdGfjPF

[root@GRPI-OCP-PV00 acmac-active]# oc get secret acm-hub-backup -n
openshift-storage -o jsonpath='{.data.AWS_SECRET_ACCESS_KEY}' | base64 --decode

> qBBOUmD2Hcx/2klW53/hE56pq0VapoXl8S+jrpR7

[root@GRPI-OCP-PV00 acmac-active]# oc get obc acm-hub-backup -o yaml -n
openshift-storage | grep "bucketName:" | awk '{print $2}'


> acm-hub-backup-4936001c-049a-426a-8f77-b58365cdae2e

[root@GRPI-OCP-PV00 acmac-active]# oc get route -n openshift-storage | grep
"s3-openshift" | awk '{print $2}'


> https://s3-openshift-storage.apps.ocp-acmac.cariprpc.it
```

5. Configurare Secret con credenziali del Bucket S3 Sorgente per OADP Operator

È necessario creare all'interno del namespace **open-cluster-management-backup** la secret **cloud-credentials** contenente i riferimenti per **access-secret** e **access-secret-key.**

```
Python
[root@GRPI-OCP-PV00 acmac-active]# vim s3-credential_acm-hub-backup

[default]
aws_access_key_id=5AHJv5hLEmOd7wdGfjPF
aws_secret_access_key=qBBOUmD2Hcx/2klW53/hE56pq0VapoXl8S+jrpR7

[root@GRPI-OCP-PV00 ocp-acmac]# oc create secret generic cloud-credentials -n
open-cluster-management-backup --from-file cloud=./s3-credential_acm-hub-backup
```

6. A questo punto, è possibile creare il **DataProtectionApplication** utilizzando il seguente template:

```
Python
--- Creazione oggetto DataProtectionApplication ---

[root@GRPI-OCP-PV00 acmac-active]# vim dpa-active.yaml

apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-acm-active
  namespace: open-cluster-management-backup
spec:
  backupLocations:
    - velero:
        config:
          insecureSkipTLSVerify: 'true'
          profile: default
          region: default
          s3ForcePathStyle: 'true'
          s3Url: 'https://s3-openshift-storage.apps.ocp-acmac.cariprpc.it'
        credential:
          key: cloud
          name: cloud-credentials
        default: true
        objectStorage:
          bucket: acm-hub-backup-4936001c-049a-426a-8f77-b58365cdae2e
          prefix: prod
        provider: aws
  configuration:
    restic:
      enable: true
    velero:
      defaultPlugins:
        - aws
        - openshift

[root@GRPI-OCP-PV00 acmac-active]# oc apply -f dpa-active.yaml
```

Il **DataProtectionApplication** contiene riferimenti per:

- Il tipo di S3 utilizzato
- Il nome del bucket S3
- Il segreto per l'autenticazione al bucket S3
- Il prefisso con cui verranno creati i backup

Una volta completata questa fase, è possibile connettersi alla dashboard di OCP e verificare che all'interno dell'operatore OADP, nella sezione **BackupStorageLocation**, lo stato dell'oggetto creato sia **Available**. Oppure da linea di comando eseguire:

```
Python
--- Check BuckupStorageLocation ---

[root@GRPI-OCP-PV00 acmac-active]# oc get buckupstoragelocation -n
open-cluster-management-backup
```

Per completare la configurazione del cluster active, è necessario creare un **BackupSchedule** e verificare che i backup siano stati creati correttamente:

```
Python
--- Definizione BuckupSchedule ---

[root@GRPI-OCP-PV00 acmac-active]# vim backupschedule.yaml

apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm
  namespace: open-cluster-management-backup
spec:
  veleroSchedule: 0 */12 * * *
  veleroTtl: 360h
  useManagedServiceAccount: true

[root@GRPI-OCP-PV00 acmac-active]# oc apply -f backupschedule.yaml
[root@GRPI-OCP-PV00 acmac-active]# oc get backup -n open-cluster-management-backup

NAME                                                 AGE
acm-credentials-schedule-20240514000025              7h23m
acm-managed-clusters-schedule-2024051400002520h      7h23m
acm-resources-generic-schedule-20240514000025        7h23m
acm-resources-schedule-20240514000025                7h23m
acm-validation-policy-schedule-20240514000025        7h23m
```

3.5. Configurazione Rozzano come passive cluster

I backup sul cluster HUB passive di Rozzano vengono importati tramite l'operator OADP.
Per configurarlo, segui questi passaggi:

1. Dalla dashboard, seleziona l'operatore ACM.
2. Vai su **MultiClusterHub**.
3. Modifica l'istanza esistente, abilitando il backup del cluster.

```
Python
  components:
    - enabled: true
      name: cluster-backup
    ...
```

In alternativa da linea di comando eseguendo:

```
Python
[root@GRPI-OCP-PV00 ~]# ocp-acmrz
[root@GRPI-OCP-PV00 ~]# oc edit multiclusterhub -n open-cluster-management

[...]
spec:
  availabilityConfig: High
  enableClusterBackup: false
  ingress:
    sslCiphers:
      - ECDHE-ECDSA-AES256-GCM-SHA384
      - ECDHE-RSA-AES256-GCM-SHA384
      - ECDHE-ECDSA-AES128-GCM-SHA256
      - ECDHE-RSA-AES128-GCM-SHA256
  overrides:
    components:
     [...]
      - enabled: true
        name: cluster-backup
     [...]
```

Dopo alcuni minuti, è possibile verificare nell'elenco degli operatori installati che:

- L'operator **OADP** esiste.
- È installato all'interno del namespace: **open-cluster-management-backup**.

```
Python
[root@GRPI-OCP-PV00 ~]# oc get subscriptions.operators.coreos.com  -n
open-cluster-management
```

```
NAME                            PACKAGE                         SOURCE
CHANNEL
advanced-cluster-management     advanced-cluster-management     redhat-operators
release-2.10
```

È necessario creare all'interno del namespace **open-cluster-management-backup** la secret **cloud-credentials** contenente i riferimenti a **access-secret** e **access-secret-key**. Recuperare le credenziali di accesso al Bucket come mostrato nel paragrafo "**Configurazione bucket S3"**:

```
Python
[root@GRPI-OCP-PV00 ~]# cd cluster-acm-dr/acmrz-passive

[root@GRPI-OCP-PV00 acmrz-passive]# oc get secret acm-hub-backup -n
openshift-storage -o jsonpath='{.data.AWS_ACCESS_KEY_ID}' | base64 --decode
> VWRhDb0yUpyPCif3Y3HC

[root@GRPI-OCP-PV00 acmrz-passive]# oc get secret acm-hub-backup -n
openshift-storage -o jsonpath='{.data.AWS_SECRET_ACCESS_KEY}' | base64 --decode
> NinP3SSOdBEt6nnyWgxcsgIJ12037jI9gCmPtl1x

[root@GRPI-OCP-PV00 acmrz-passive]# oc get obc acm-hub-backup -o yaml -n
openshift-storage | grep "bucketName:"
> acm-hub-backup-5d0ad6e3-1c9d-4792-bd64-6207cd21c3a8

[root@GRPI-OCP-PV00 acmrz-passive]# oc get route -n openshift-storage | grep
"s3-openshift"
> https://s3-openshift-storage.apps.ocp-acmrz.cariprpc.it
```

Creare la secret per OADP Operator:

```
Python
[root@GRPI-OCP-PV00 acmrz-passive]# vim s3-credential_acm-hub-backup

[default]
aws_access_key_id=VWRhDb0yUpyPCif3Y3HC
aws_secret_access_key=NinP3SSOdBEt6nnyWgxcsgIJ12037jI9gCmPtl1x

[root@GRPI-OCP-PV00 ocp-acmrz]# oc create secret generic cloud-credentials -n
open-cluster-management-backup --from-file cloud=./s3-credential_acm-hub-backup
```

A questo punto, è possibile creare il **DataProtectionApplication** utilizzando il seguente template:

```
Python
--- Creazione oggetto ObjectBucketClaim sorgente ---

[root@GRPI-OCP-PV00 acmrz-passive]# vim dpa-passive.yaml

apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-acm-passive
  namespace: open-cluster-management-backup
spec:
  backupLocations:
    - velero:
        config:
          insecureSkipTLSVerify: 'true'
          profile: default
          region: default
          s3ForcePathStyle: 'true'
          s3Url: 'https://s3-openshift-storage.apps.ocp-acmrz.cariprpc.it'
        credential:
          key: cloud
          name: cloud-credentials
        default: true
        objectStorage:
          bucket: acm-hub-backup-5d0ad6e3-1c9d-4792-bd64-6207cd21c3a8
          prefix: prod
        provider: aws
  configuration:
    restic:
      enable: true
    velero:
      defaultPlugins:
        - aws
        - openshift

[root@GRPI-OCP-PV00 acmrz-passive]# oc apply -f dpa-passive.yaml
```

Il **DataProtectionApplication** contiene riferimenti per:

- Il tipo di S3 utilizzato
- Il nome del bucket S3
- Il segreto per l'autenticazione al bucket S3
- Il prefisso con cui verranno creati i backup

Una volta completata questa fase, è possibile connettersi alla dashboard di OCP e verificare che all'interno dell'operatore OADP, nella sezione **BackupStorageLocation**, lo stato dell'oggetto creato sia **Available**. Oppure da linea di comando eseguire:

```
Python
--- Check BuckupStorageLocation ---
```

```
[root@GRPI-OCP-PV00 acmrz-passive]# oc get buckupstoragelocation -n
open-cluster-management-backup

NAME                PHASE
dpa-acm-passive     Available
```

Per completare la configurazione del cluster di Rozzano come passivo, è necessario creare l'oggetto **Restore** utilizzando il seguente template e verificare che il ripristino termini senza errori:

```
Python
--- Crezione oggetto Restore ---
[root@GRPI-OCP-PV00 acmrz-passive]# vim restoreschedule.yaml

apiVersion: cluster.open-cluster-management.io/v1beta1
kind: Restore
metadata:
  name: restore-acm-passive-sync
  namespace: open-cluster-management-backup
spec:
  cleanupBeforeRestore: CleanupRestored
  restoreSyncInterval: 60m
  syncRestoreWithNewBackups: true
  veleroCredentialsBackupName: latest
  veleroManagedClustersBackupName: skip
  veleroResourcesBackupName: latest

[root@GRPI-OCP-PV00 acmrz-passive]# oc apply -f restoreschedule.yaml
[root@GRPI-OCP-PV00 acmrz-passive]# oc get restore -n
open-cluster-management-backup

PHASE         MESSAGE
Enabled       Velero restore have run to completion, resto will continue to sync
              with new backups

[root@GRPI-OCP-PV00 acmrz-passive]# oc get backup -n open-cluster-management-backup

NAME                                                 AGE
acm-credentials-schedule-20240514000025              7h23m
acm-managed-clusters-schedule-2024051400002520h      7h23m
acm-resources-generic-schedule-20240514000025        7h23m
acm-resources-schedule-20240514000025                7h23m
acm-validation-policy-schedule-20240514000025        7h23m
```

4. Procedura di Failover

La procedura di **failover** viene eseguita quando il cluster HUB active non è più raggiungibile.

Per simulare questo scenario, il cluster HUB Acilia è stato spento. Tuttavia, poiché si tratta di un test di failover che successivamente coinvolgerà un test di failback, sono state eseguite le seguenti operazioni:

1. Distruzione del cluster di Acilia
2. Patch sull'oggetto **Restores** di Rozzano

Una volta spento il cluster Acilia, è possibile eseguire il seguente comando, che avvierà il ripristino dei dei dati attivi e l'import dei cluster managed:

```
Python
    oc patch restores.cluster.open-cluster-management.io/restore-acm-passive-sync -p
        '{"spec":{"veleroManagedClustersBackupName":"latest"}}' -n
        open-cluster-management-backup --type merge
```

Dopo alcuni minuti, sulla dashboard di ACM del cluster HUB di Rozzano, i cluster managed appariranno nello stato "**Ready**".

Una volta verificato che tutti i cluster gestiti sono nello stato "**Ready**" e che tutte le policy applicate ai cluster sono in uno stato consistente, è possibile completare la procedura di failover seguendo questi passaggi:

1. Cancellazione del **Restore** task:

```
Python
[root@GRPI-OCP-PV00 acmrz-passive]# ocp-acmrz
[root@GRPI-OCP-PV00 acmrz-passive]# oc delete -n open-cluster-management-backup
restore-acm-passive-sync
```

2. Creazione del **BackupSchedule** per avviare i nuovi backup dal corrente cluster Attivo (Rozzano) e verifica dell'esecuzione:

```
Python
--- Definizione BuckupSchedule ---

[root@GRPI-OCP-PV00 acmrz-passive]# vim backupschedule.yaml

apiVersion: cluster.open-cluster-management.io/v1beta1
kind: BackupSchedule
metadata:
  name: schedule-acm
  namespace: open-cluster-management-backup
spec:
  veleroSchedule: 0 */12 * * *
  veleroTtl: 360h
```

```
[root@GRPI-OCP-PV00 acmrz-passive]# oc apply -f backupschedule.yaml
[root@GRPI-OCP-PV00 acmrz-passive]# oc get backup -n open-cluster-management-backup

NAME                                                 AGE
acm-credentials-schedule-20240514000025              7h23m
acm-managed-clusters-schedule-2024051400002520h      7h23m
acm-resources-generic-schedule-20240514000025        7h23m
acm-resources-schedule-20240514000025                7h23m
acm-validation-policy-schedule-20240514000025        7h23m
```

5. Procedura di Failback

La procedura di **failback** comporta la ripetizione delle stesse operazioni di configurazioni eseguite per il **failover**, invertendo il cluster Active con quello Passive, con la precauzione di eseguire un'attività di pulizia dei dati che fanno riferimento al precedente cluster active.

Ecco l'elenco delle operazioni da eseguire sul vecchio cluster Active di Acilia (una volta riavviato) e sul cluster Rozzano:

1. Ripristino cluster Acilia.
2. Eseguire tutti i passaggi documentati nel capitolo Configurazione Active Passive a parti invertite.
3. Creare il task di ripristino come documentato nella sezione di configurazione per il cluster in modalità passiva su Acilia ed attendere replica dei dati su S3.
4. Distruggere cluster Rozzano.
5. Eseguire la procedura di failover su questo nuovo cluster (da Rozzano ad Acilia).
6. Configurare Acilia come active cluster e creare il **BackupSchedule**.
7. Effettuare il ripristino del cluster di Rozzano e configurarlo come passive cluster.

WARNING: **Così come fatto durante i test, anche in una vera situazione di DR (Disaster Recovery)** è necessario partire sempre da una nuova installazione di ACM da configurare come cluster passivo.