WORKSTATION SETUP

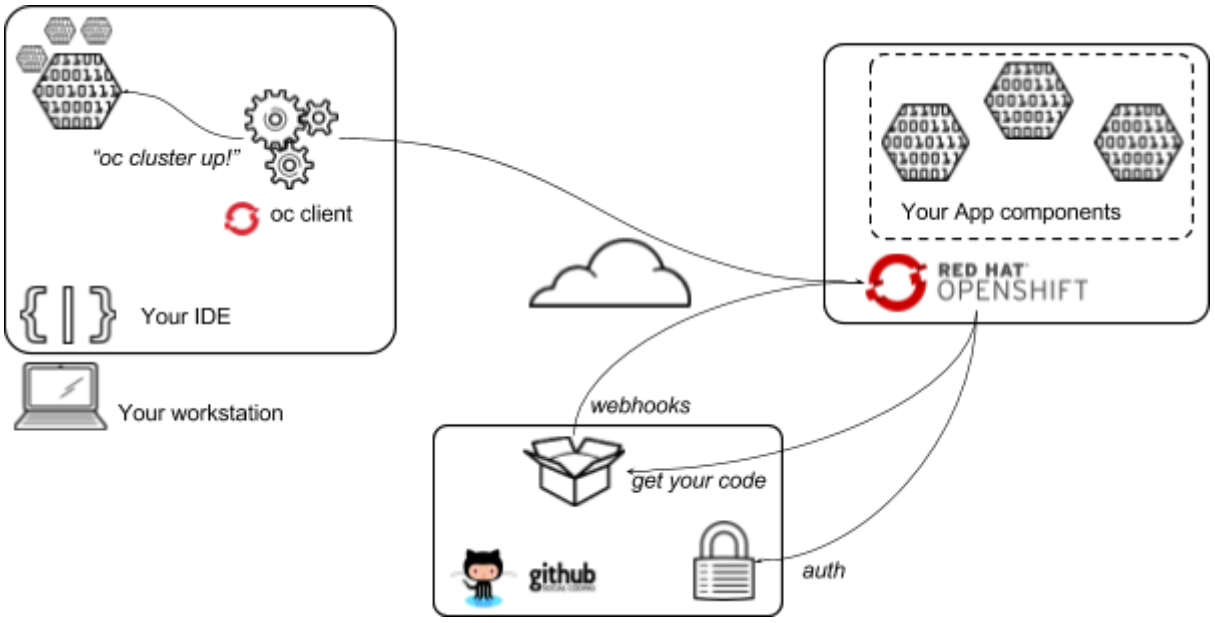
Questa guida illustra come predisporre una postazione di sviluppo per OpenShift.

Con openshift si hanno molti modi per interagire e sviluppare applicazioni containerizzate senza dover diventare degli esperti docker o amministratori di sistema!

In questa guida illustriamo come preparare due workstation, la prima leggera ed equipaggiata solo con i client vi permetterà di sviluppare e rilasciare applicazioni direttamente sull'infrastruttura in cloud la seconda vi consentirà di eseguire anche un cluster openshift localmente.

Come IDE di sviluppo potete utilizzarne uno a vostra scelta; JBoss Developer Studio vi semplificherà un po' il lavoro perchè Red Hat distribuisce i plugin per controllare le applicazioni (build, deploy, ecc...) direttamente dal vostro ambiente di sviluppo.

Le componenti principali in gioco sono rappresentate nella grafica che segue:



Componente	Descrizione	Versione	Note
docker	il linux container engine utilizzato da openshift per eseguire le vostre componenti applicative		L'engine è necessario sulla workstation **solo** nel caso in cui si voglia eseguire il cluster OpenShift localmente. Per workstation windows e apple, docker non è nativo; verrà quindi eseguuita una virtual machine con kernel linux

			a cui i client docker si collegano. Tenete presente di questo aspetto nei paragrafi che seguono.
oc	client a riga di comando per il collegamento ad un cluster OpenShift e esecuzione di un cluster locale		
oadm	client di amministrazione del cluster		
JBoss Developer Studio	IDE di sviluppo supportato da Red Hat (basato su Eclipse)		Link per il download: https://developers.redhat.com/products/devstudio/overview/

Per l'hackathon il team Red Hat ha predisposto una infrastruttura OpenShift in alta affidabilità su IaaS Amazon (AWS).

Visto che il tempo è prezioso! per semplificarvi il lavoro l'infrastruttura in OpenShift è equipaggiata con:

- numerose **immagini** (application server, code, web server, ecc...) che potete utilizzare per realizzare le vostre componenti
- **storage** che potrete sfruttare per le vostre componenti applicative che richiedono persistenza
- **logging centralizzato** per debuggare in modo semplice le vostre applicazioni
- **metriche** per monitorare quante risorse le vostre componenti stanno consumando
- numerosi **hello world** integrati con i servizi utili agli use-case richiesti nell'hackathon
- un servizio di **proxy SPID** per semplificare l'integrazione delle vostre applicazioni con il servizio di autenticazione della PA

Di seguito trovate alcuni dettagli:

OpenShift WEB Console	https://openshift-master.justcodeon.it/console
SPID di test su OpensShift	https://identityserver-spidtest.apps.justcodeon.it https://backoffice-spidtest.apps.justcodeon.it

Il cluster AWS è stato configurato in single-sign-on con **github**, pertanto per accedere potete utilizzare il vostro account github dopo averlo comunicato qui al team Red Hat per avere le autorizzazioni.

Appena ricevete l'ok potete accedere alla GUI Web di Openshift e potete iniziare a preparare le applicazioni.

Se avete già un repo github pronto potete provare direttamente la vostra app scegliendo una delle immagini disponibili (java, php, nodejs, mongo, ecc...), aggiungere il riferimento al vostro repo e OpenShift farà il resto (compilazione, creazione del container, deploy e binding dei servizi).
Potete chiedere al mentor di illustrarvi questa procedura se avete difficoltà.

Setup di una Workstation base

Questa configurazione prevede l'utilizzo di:

- una vostra postazione di sviluppo equipaggiata con
    - un browser
    - i client oc per accedere al cluster via cli
- un account github

e opzionalmente:

- un repo github (nel caso si voglia sfruttare il source 2 image)
- l'utilizzo di un vostro IDE per il coding e il commit sul repo oppure uno già integrato con il cluster openshift (vi darà alcune utility di gestione via ide delle vostre applicazioni):
    - JBoss Developer Studio
    - Eclipse + ocp plugin
    - IntelliJ più oc da terminale

OPENSHIFT ACCOUNT SETUP

Per un pieno controllo del vostro ambiente, installate la cli (il client OC) nel seguito

OC CLIENT SETUP

1. Seguire le istruzioni indicate per il download nella pagina https://openshift-master.justcodeon.it/console/command-line

workstation type	package	note
linux	oc-3.6.173.0.5-linux.tar.gz	
windows	oc-3.6.173.0.5-windows.zip	
mac	oc-3.6.173.0.5-macosx.tar.gz	

2. scompattare il package

3. spostare l'eseguibile oc dentro un default path o aggiungere ai "default path" la posizione del file oc
4. connettersi al cluster su AWS

BASH COMPLETION

[workstation linux]

La bash completion è un'utility di autocompletamento delle opzioni annidate in un comando di shell.
La community di openshift manutiene i parser per il client oc, ecco come installarli:

yum install -y bash-completion &&\
curl -G https://raw.githubusercontent.com/openshift/origin/master/contrib/completions/bash/oc > /etc/bash_completion.d/oc &&\
curl -G https://raw.githubusercontent.com/openshift/origin/master/contrib/completions/bash/oadm > /etc/bash_completion.d/oadm

LOGIN AL CLUSTER VIA OC CLIENT

1. Accedere alla web gui per prelevare il token di autenticazione: https://openshift-master.justcodeon.it/console/command-line
2. copiare il token contenuto del box <token>
3. dalla shell della tua workstation lancia il comando:
   oc login https://openshift-master.justcodeon.it --token=<token>
4. fare un test per verificare la sessione:
   oc get pods

Workstation full-stack

Questa configurazione vi permette di eseguire openshift sulla workstation: sostanzialmente oc, sfruttando una versione containerizzata di openshift, avvia un cluster locale
Lo scopo di questo mini-cluster è relativo al solo sviluppo e consumerà un minimo di risorse, avrete un po' di autonomia e potrete acquisire un pò di nozioni di amministrazione del sistema.

Vediamo come preparare la workstation:

INSTALLAZIONE DOCKER

TUNING DOCKER CONFIG

vim /etc/sysconfig/docker
OPTIONS='--selinux-enabled --insecure-registry 172.30.0.0/16 --log-driver=journald
--insecure-registry registry.access.redhat.com'
ADD_REGISTRY='--add-registry registry.access.redhat.com'

OC CLUSTER UP

il comando “oc cluster up” esgue un cluster locale e containerizzato di openshift origin.
Il cluster può essere eseguito in due modalità:

- effimero: allo shutdown del cluster, i dati (container, immagini, template, applicazioni, deplyment, ecc...) vengono eliminati
- persistente: sul filesystem della workstation, il cluster mantiene configurazioni e oggetti preparati

Nota: in caso di cluster persistente, il comando per il primo avvio è differente dai successivi.
Di seguito sono riportati alcuni dei parametri potenzialmente utili per la configurazione del cluster locale

FIRST TIME CLUSTER UP

```
oc cluster up \
  --http-proxy=http://proxy.istat.it:3128/ \
  --https-proxy=http://proxy.istat.it:3128/ \

--no-proxy='test-auth-rh7,test-auth-rh7.istat.it,docker-registry.default.svc.cluster.local,10
.0.0.0/8,172.30.1.1,172.30.0.0/16,172.0.0.0/8,192.168.0.0/16' \
  --env 'HTTP_PROXY=http://proxy.istat.it:3128/' \
  --env 'HTTPS_PROXY=http://proxy.istat.it:3128/' \
  --env
'"NO_PROXY=test-auth-rh7,test-auth-rh7.istat.it,docker-registry.default.svc.cluster.local,
10.0.0.0/8,172.30.1.1,172.30.0.0/16,172.0.0.0/8,192.168.0.0/16"' \
  --host-data-dir="/var/ocp_data/host-data" \
  --host-pv-dir="/var/ocp_data/host-pv" \
  --host-volumes-dir="/var/ocp_data/host-volumes" \
  --logging=false \
  --metrics=false \
  --public-hostname="test-auth-rh7.istat.it" \
  --routing-suffix="cloud.istat.it" \
  test-auth-rh7
```

STOP THE CLUSTER

```
oc cluster down
```

NEXT CLUSTER START THE CLUSTER

```
oc cluster up \
  --http-proxy=http://proxy.istat.it:3128/ \
  --https-proxy=http://proxy.istat.it:3128/ \

--no-proxy='test-auth-rh7,test-auth-rh7.istat.it,docker-registry.default.svc.cluster.local,10
.0.0.0/8,172.30.1.1,172.30.0.0/16,172.0.0.0/8,192.168.0.0/16' \
  --env 'HTTP_PROXY=http://proxy.istat.it:3128/' \
  --env 'HTTPS_PROXY=http://proxy.istat.it:3128/' \
  --env
'"NO_PROXY=test-auth-rh7,test-auth-rh7.istat.it,docker-registry.default.svc.cluster.local,
10.0.0.0/8,172.30.1.1,172.30.0.0/16,172.0.0.0/8,192.168.0.0/16"' \
  --host-data-dir="/var/ocp_data/host-data" \
  --host-pv-dir="/var/ocp_data/host-pv" \
  --host-volumes-dir="/var/ocp_data/host-volumes" \
  --logging=false \
  --metrics=false \
  --public-hostname="test-auth-rh7.istat.it" \
  --routing-suffix="cloud.istat.it" \
  --use-existing-config test-auth-rh7
```

ACCESS THE CLUSTER CONTAINER

```
docker exec -it origin bash
```

ACCESS THE SERVER

```
oc login https://test-auth-rh7.istat.it:8443
```

CREATE TEMPLATES

```
mkdir /root/ocp_utils/
cd /root/ocp_utils/; git clone https://github.com/jboss-openshift/application-templates
cd /root/ocp_utils/application-templates/; oc create -f ./jboss-image-streams.json -n
openshift

mkdir /root/ocp_utils/application-templates-fuse
cd /root/ocp_utils/application-templates-fuse/; git clone
https://github.com/jboss-fuse/application-templates
cd /root/ocp_utils/application-templates-fuse/application-templates; oc create -f
./fis-image-streams.json -n openshift
for i in /root/ocp_utils/application-templates-fuse/application-templates/quickstarts/; do
oc create -f $i -n openshift; done
```

ACCESS THE REGISTRY & PUSHING IMAGES

oc login -u system:admin
oc get svc docker-registry -n default
oc login -u developer

To push arbitrary images from your local Docker daemon to this registry:

Obtain your OpenShift token and store it in a variable
OPENSHIFT_TOKEN=$(oc whoami -t)

Login to the registry
docker login -u developer -p ${OPENSHIFT_TOKEN} 172.30.1.1:5000

Tag an image (format <REGISTRY_IP>:5000/<PROJECT>/name[:TAG]) with your registry and namespace, and push it
docker pull nginx:latest
docker tag nginx:latest 172.30.1.1:5000/myproject/nginx:latest
docker push 172.30.1.1:5000/myproject/nginx:latest

ISSUES

IMAGES NOT FOUND FOR TAGS ERROR

Not always default images are published for the whole stack.. you can try to retag latest images after manual download:
docker pull registry.access.redhat.com/openshift3/**ose-metrics-heapster**
docker images
docker tag **cc6454e9d765**
registry.access.redhat.com/openshift3/ose-metrics-heapster:**v3.5.5.15**

RIFERIMENTI

OC CLUSTER UP/DOWN

ref**:** https://github.com/openshift/origin/blob/master/docs/cluster_up_down.md
ref: https://developer.fedoraproject.org/deployment/openshift/about.html
ref: http://www.colliernotes.com/2016/08/openshift-cluster-up-on-fedora.html