

Déployez une appli sur Azure avec Docker, Terraform et Jenkins

Obab Aldbiyat

06/16/2023

Liens nécessaires pour le projet :

Dockerfile : <https://github.com/obabaldbiyat/brief14.git>

Projet : <https://github.com/obabaldbiyat/JenkinsDevOpsAppAzureDockerTerraform.git>

On commence sur Jenkins à créer un “nouveau projet > pipeline” et ensuite la construction de notre Jenkinsfile avec ces cinq étapes principales :

- 1- Clone, Build, Tag and Push de l'image Docker sur DockerHub.
- 2- Connection sur notre compte Azure.
- 3- Construction du projet Terraform pour le Staging Environnement avec l'image docker créée
- 4- La vérification du fonctionnement de notre AppWeb
- 5- Continuer la construction de projet Terraform pour la Prod Environnement avec l'image docker créée.

Paramètres à faire avant

Installation sur la VM principale :

- Jenkins
- Docker
- Terraform

Commandes à exécuter sur la VM, suite à l'installation de Jenkins et Docker :

- Docker:
`$ sudo usermod -aG docker $USER`
`$ sudo chown $USER:docker /var/run/docker.sock`

```
$ sudo chmod 666 /var/run/docker.sock
```

- Jenkins :

<https://stackoverflow.com/questions/11880070/how-to-run-a-script-as-root-in-jenkins>

Plugin à télécharger sur Jenkins

- Docker Pipeline
- Docker Commons Plugin
- Azure AD Plugin
- Azure Credentials
- Azure SDK API Plugin
- Terraform Plugin
- Git

Jenkinsfile

1- Pull, Build and Push de l'image Docker sur DockerHub

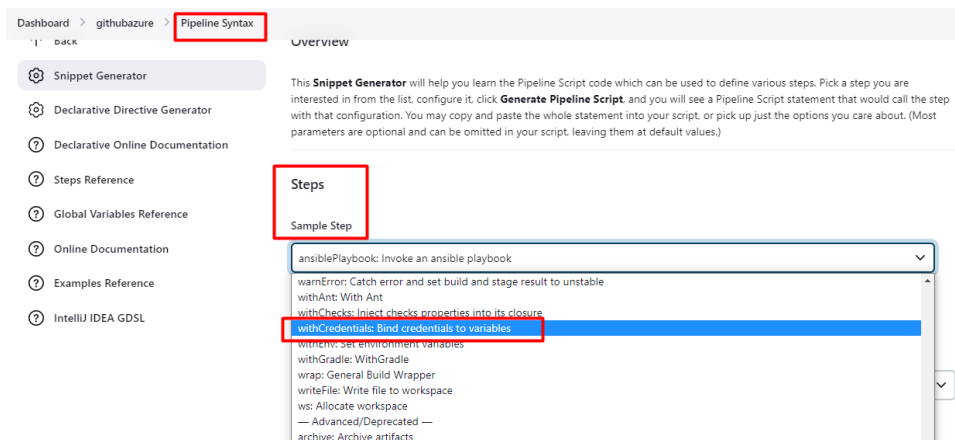
On commence le projet par “git clone” de l'app de notre repository GitHub, en utilisant les “credentialsId” déjà paramétrer à l'aide de “Pipeline Syntax” de Jenkins, pour se connecter sur GitHub, puis, l'URL de notre repository.

<https://github.com/obabaldbiyat/brief14>

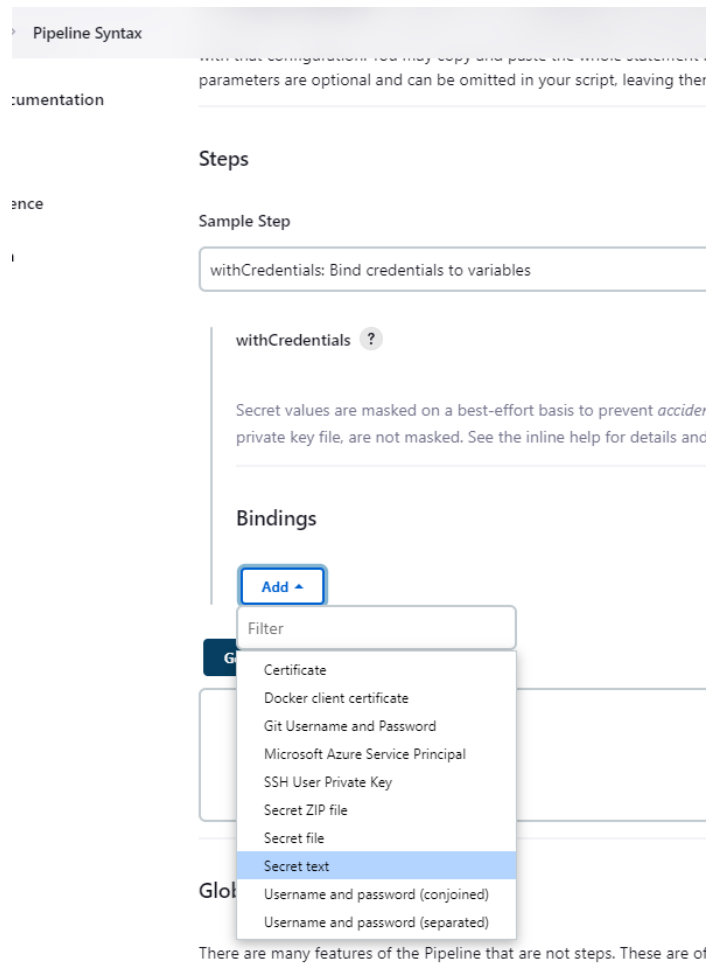
Ensuite, on build l'image, tag l'image, se connecter à DockerHub avec le “credentialsId” déjà paramétré également, et là on push l'image sur notre DockerHub.

Etape de paramétrage les “credentialsId”:

1- Choisir le step > credentials to variables



2- Choisir Add > Secret text



3- Déclarer le nom de variable et choisir Add Credentials

Bindings

≡ Secret text ?

Variable ?

DockerhubPWD

Credentials ?

Add ▲

Jenkins

Add ▼

4- Déclarer le nom et le mot de passe

Global (Jenkins, nodes, items, all child items, etc.)

Username ?

obab

☐ Treat username as secret ?

Password ?

ID ?

Description ?

Add Cancel

5- Sortie le code à utiliser dans Jenkinsfile.

Bindings

Secret text ?

Variable ?

DockerhubPWD

Credentials ?

Add ▲

Jenkins

Add ▼

Generate Pipeline Script

```
withCredentials([string(credentialsId: '', variable: 'DockerhubPWD')]) {
    // some block
}
```

2- Connection sur notre compte Azure.

Sur le terminal d'Azure Cloud Shell, on execute la commande suivant pour récupérer l'identifiant, le tenant et le code sucré à utiliser pour Jenkins :

```
az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/***** Subscription ID
*****
```

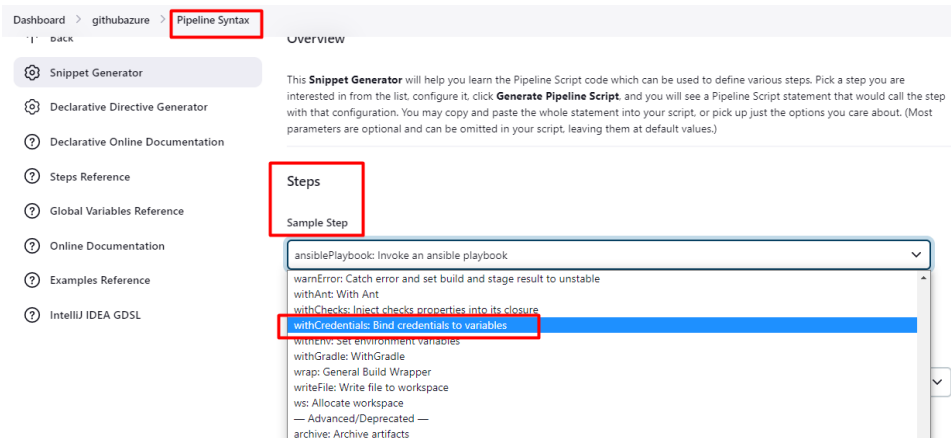
Sortie à copier et à enregistrer :

```
"appId": "*****",
"displayName": "*****",
"password": "*****",
"tenant": "*****"
```

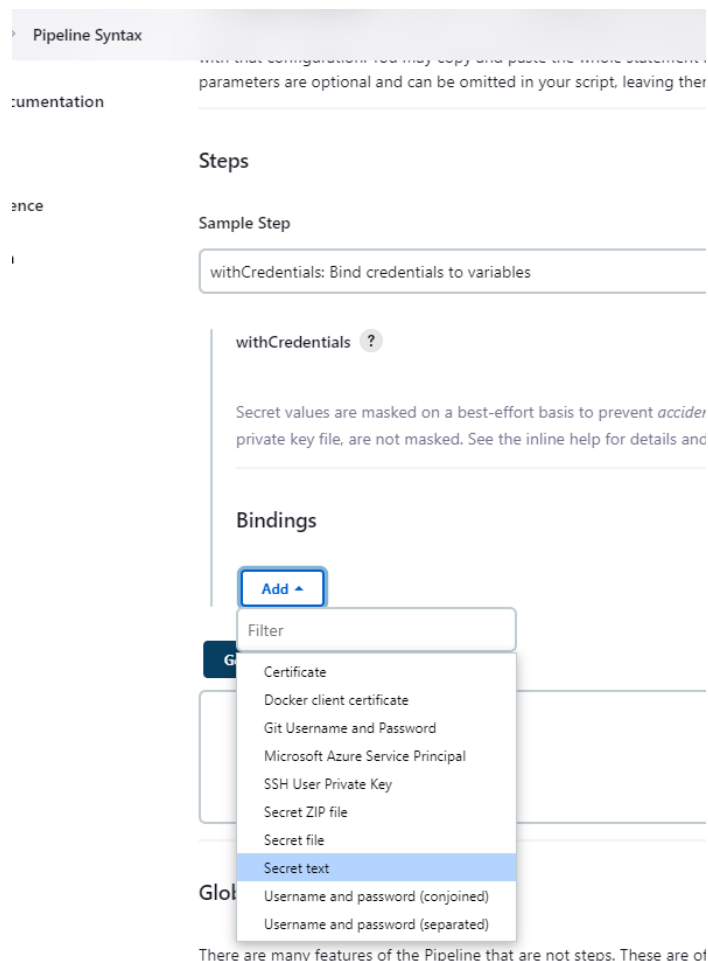
On utilise ces informations dans le “Pipeline Syntax” également pour paramétrer des variables qui vont nous permettre de se connecter à Azure :

Les étapes pour configurer les variables à l’aide du plugin Azure Service Principal :

1- Choisir le step > credentials to variables



2- Choisir Add > Secret text



3- Choisir Azure Service Principal

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain
Global credentials (unrestricted)

Kind
Username with password
Username with password
Azure Key Vault Secret
Azure Key Vault Secret Certificate
Azure Managed Identity
Azure Service Principal
GitHub App
OpenShift Username and Password
SSH Username with private key
Secret file
Secret text
X.509 Client Certificate
Certificate

☐ Treat username as secret ?

Password ?

4- Remplir les paramètres avec les données récupéré suite à notre commande sur Azure Cloud Shell.

Azure Service Principal

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Subscription ID ?

Client ID ?

Client Secret ?

Or, Certificate ?
--- Select a Certificate ---

+ Add

Tenant ID ?

3- Terraform

On a organisé le projet Terraform en trois dossiers sur GitHub :

<https://github.com/obabaldbiyat/JenkinsDevOpsAppAzureDockerTerraform/tree/main>

ProdEnvironment (dossier root avec un main.tf pour chercher le module Childe et un scripte pour l'installation de docker et builder l'image de l'app)

StagingEnvironment (dossier root avec un main.tf pour chercher le module Childe et un scripte pour l'installation de docker et Builder l'image de l'app)

WebserverModule le dossier Childe qui va déployer l'infrastructure pour l'environnement demandé.

Des Outputs dans chaque dossier pour nous fournir les données nécessaires à trouver le groupe de ressource et les adresses IP.

- On commence ici avec la déclaration de nos variables sur Jenkins pour se connecter sur Azure, avec le texte suivant :

```
pipeline {  
    agent any  
    environment {  
        MY_CRED = credentials('*****credentials ID*****')  
    }  
}
```

- Ensuite, deux stages : le premier pour se connecter sur github et récupérer le repository demandé et le deuxième pour se connecter à Azure avec les variables paramétrées.

- Déploiement du projet Terraform pour chaque l'environnement.

On utilise la commande (`sh 'cd ***l'environnement demandé**'`) pour demander à Terraform quel environnement à déployer.

4- La vérification du fonctionnement de notre AppWeb

On a ajouté le stage 'Sanity check' entre le déploiement de deux environnements pour demander si le fonctionnement de l'app sur le Staging est "OK" pour continuer le Building vers la Prod:

```
stage('Sanity check') {
```



```

steps {
    input "Does the staging environment look ok?"
}
}

```

Puis, si c'est OK, le déploiement continue de l'environnement de la Prod avec l'App en Docker.

Monitoring et les Alerts:

Sur le portail Azure > VM de la Prod > on configure le monitoring de la VM et les alerts qu'on a besoin, comme l'utilisation du CPU, le Mémoire Ram et les Inbounds:

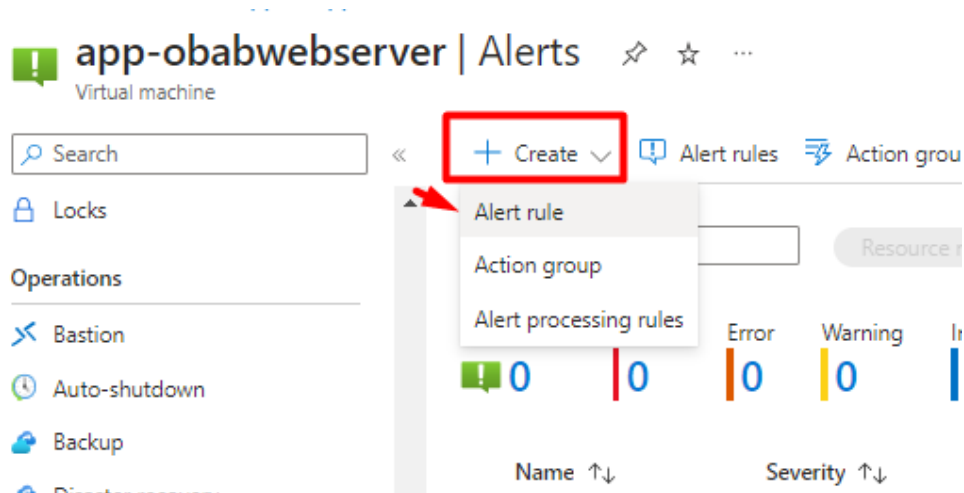
1- VM > Monitorin >

Insights > Enable

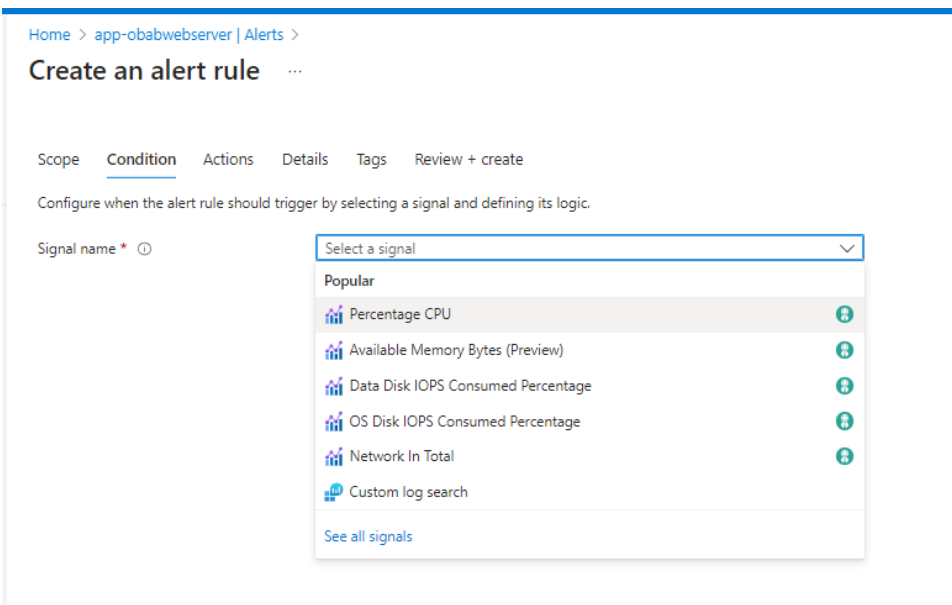
Alerts > Add

The screenshot shows the Azure portal interface for a virtual machine named 'app-obabwebserver'. The left sidebar contains a navigation menu with categories like Operations, Monitoring, and Automation. Under the 'Monitoring' section, 'Insights' and 'Alerts' are highlighted with red boxes. The main content area displays the 'Essentials' tab for the VM, showing details such as the resource group 'OBAB_PROD_APP' (highlighted with a red box), status 'Running', location 'West Europe', and subscription 'Simplon ASA Lyon Admin Cloud P22'. Below this, the 'Properties' tab is active, showing a table of VM properties including computer name, operating system, image publisher, and size. The 'Networking' section shows the public IP address and network interface. The 'Size' section shows the VM size as 'Standard F2' with 2 vCPUs.

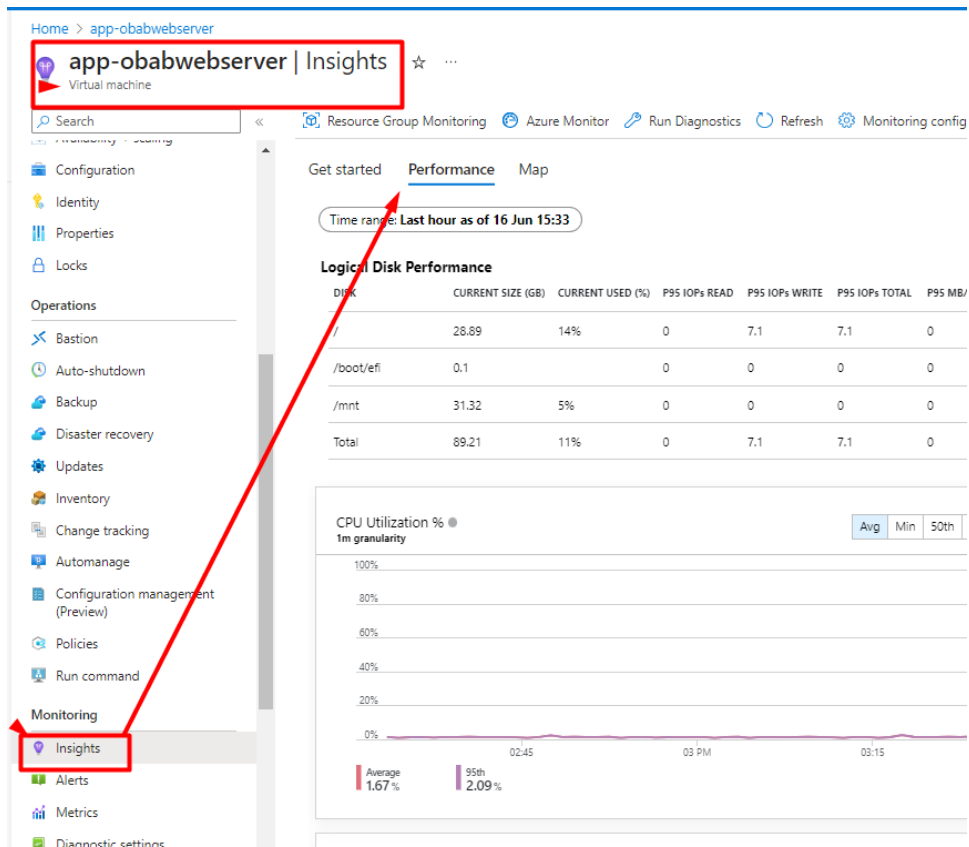
2- Création des Alerts



3- Choix des Alerts Rules



4- Insights > Performance : pour observer et analyser le fonctionnement de la VM



5- Sortie des Analyses

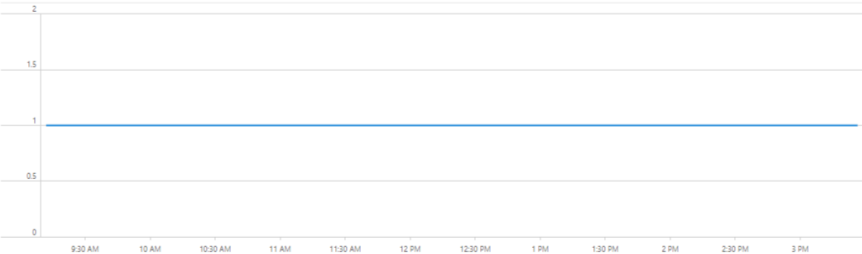
Performance Analysis

TimeRange
Last 24 hours

Trend

In the graph below a default counter (counter) has been selected, select the dropdown to choose a different counter.

Counter Aggregates
Heartbeat Average



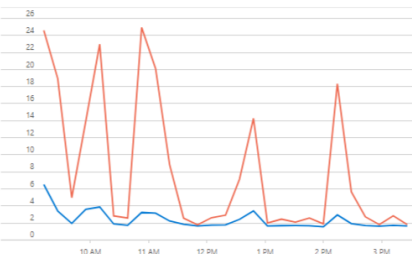
Average (Avg)
1

Performance Charts

Customize or add your own charts below in edit mode or by using the advanced editor

CPU Utilization %

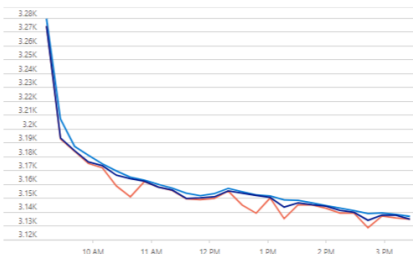
Aggregates
2 selected



Average (Avg) 2.37
P95th (Avg) 8.15

Available Memory MB

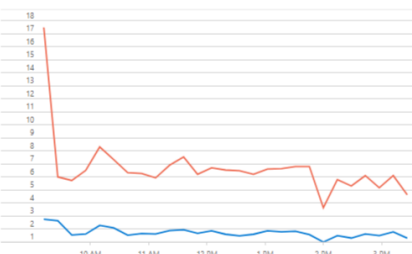
Aggregates
3 selected



Average (Avg) 3.16 K
P5th (Avg) 3.16 K
P95th (Avg) 3.16 K

Logical Disk IOPS

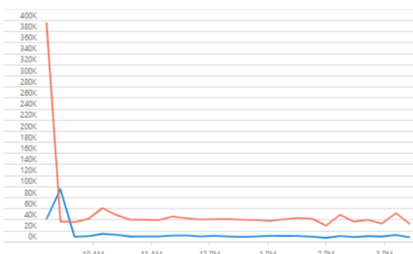
Aggregates
2 selected



Average (Avg) 1.73
P95th (Avg) 6.66

Logical Disk MB/s

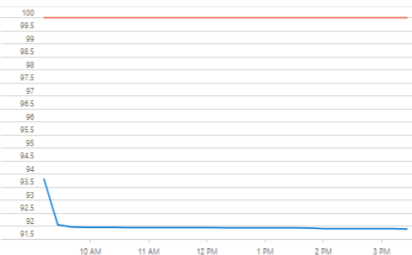
Aggregates
2 selected



Average (Avg) 14.9 K
P95th (Avg) 54.6 K

Max Logical Disk Used %

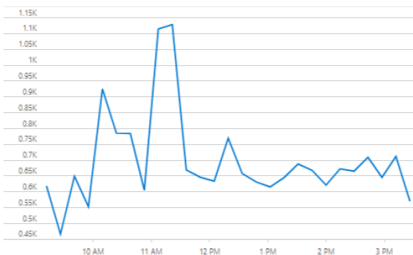
This step defines no parameters and could be removed.



Average (Avg) 93.8
P95th (Avg) 100

Bytes Sent Rate B/s

Aggregates
Average



Average (Avg) 698

Bytes Received Rate B/s

Aggregates
Average

This step defines no parameters and could be removed.