

Déployer un projet Express en utilisant Gitlab Devops

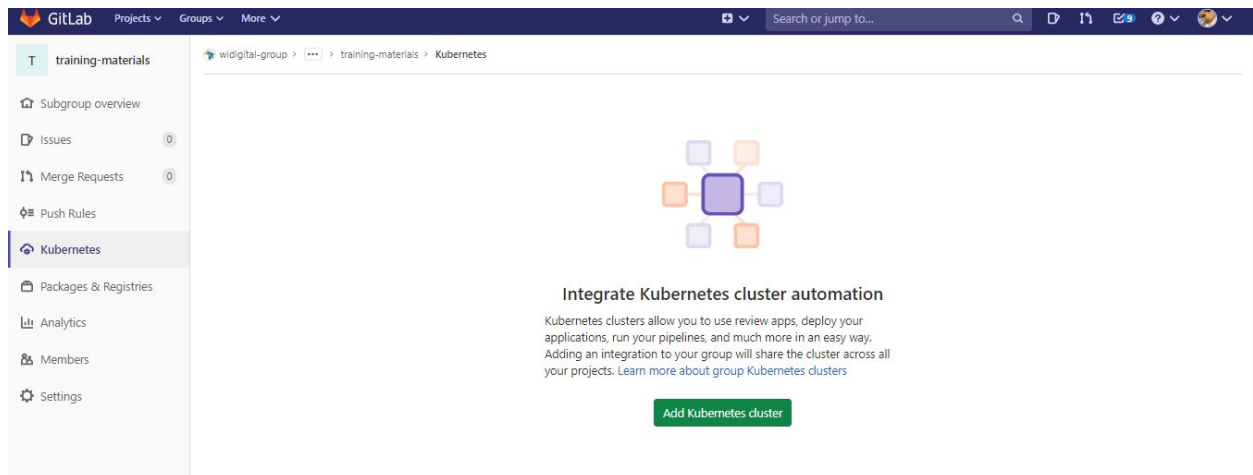
1- Ajouter un cluster kubernetes existant

Si vous disposez d'un cluster Kubernetes existant, vous pouvez l'ajouter à un projet, un groupe ou une instance. Nous allons l'ajouter à un groupe, "training-materials", créé spécialement pour ce lab.

Accédez à votre:

1. Page **Opérations** du projet > **Kubernetes** , pour un cluster au niveau du projet.
2. Page **Kubernetes** du groupe , pour un cluster au niveau du groupe.
3. Zone **d'administration**> page **Kubernetes** , pour un cluster au niveau de l'instance.

2. Cliquez sur Ajouter un cluster Kubernetes.



Cliquez sur l'onglet **Ajouter un cluster existant** et renseignez les détails:

1- Nom du cluster Kubernetes (obligatoire): Le nom que vous souhaitez donner au cluster.

2- Portée de l'environnement (obligatoire): **environnement associé** à ce cluster.

3- URL de l'API (obligatoire):

C'est l'URL que GitLab utilise pour accéder à l'API Kubernetes. Kubernetes expose plusieurs API, nous voulons l'URL «de base» qui leur est commune à toutes. Par exemple, <https://kubernetes.example.com> plutôt que <https://kubernetes.example.com/api/v1>.

Obtenez l'URL de l'API en exécutant cette commande:

```
> kubectl cluster-info | grep 'Kubernetes master' | awk '/http/ {print $NF}'
```

```
ahmedhosni_contact@cloudshell:~ (widiigital-ci)$ kubectl cluster-info | grep 'Kubernetes master' | awk '/http/ {print $NF}'  
https://23.236.53.242
```

4- Certificat CA (obligatoire):

Un certificat Kubernetes valide est nécessaire pour s'authentifier auprès du cluster. Nous utiliserons le certificat créé par défaut.

Listez les secrets avec `kubectl get secrets`, et l'un d'entre eux devrait être nommé de la même manière `default-token-xxxxx`. Copiez ce nom de jeton pour l'utiliser ci-dessous.

```
ahmedhosni_contact@cloudshell:~ (widiigital-ci)$ kubectl get secrets
```

NAME	TYPE	DATA	AGE
default-token-nsrn4	kubernetes.io/service-account-token	3	36d
esp-tls	kubernetes.io/tls	3	36d
gitlab-token-fszh2	kubernetes.io/service-account-token	3	12m
incendiary-beetle-nginx-ingress-backend-token-vm6pg	kubernetes.io/service-account-token	3	36d
incendiary-beetle-nginx-ingress-token-6k2fg	kubernetes.io/service-account-token	3	36d
sh.helm.release.v1.mycache.v1	helm.sh/release.v1	1	3d18h

Obtenez le certificat en exécutant cette commande:

```
> kubectl get secret <SECRET_NAME> -o jsonpath="{['data']['ca\.crt']}" |  
base64 --decode
```

```
ahmedhosni_contact@cloudshell:~ (widiigital-ci)$ kubectl get secret default-token-nsrn4 -o jsonpath="{['data']['ca\.crt']}" | base64 --decode  
-----BEGIN CERTIFICATE-----  
MIIDCzCCAfOgAwIBAgIQcNuS8BADt01SjXoXY4G2PTANBgqhkiG9w0BAQsFADAv  
MS0wKwYDVQQDEyQ3NDNDNhODlmNS1kZDgxLTQ3MGMTODQ0ZS1lNTNhYTZyZTBhNTUw  
HhcnMjAwNzE3MTQ1NzI2WWhcnMjUwNzE2MTU1NzI2WjAvMS0wKwYDVQQDEyQ3NDNDNh  
ODlmNS1kZDgxLTQ3MGMTODQ0ZS1lNTNhYTZyZTBhNTUwggEiMA0GCSqGSIb3DQEB  
AQUAA4IBDwAwggEKAoIBAQCpBS1egRagxOxJks0355kDtAOCdwQjORqSySVj0IH  
YgsG/8dMwggSvMDK6q8stdBdgUTFWegqcdWu0/4CrtiL98y21UZKYj7+7KjGMKE  
OLaKhWUYAES0zncRNgtkiFNq2HW+FsmDPRIB3LZ8b3ht8HMKpo95Nat7F3xb+n1A  
R+M5z3NIWJmXcd6h4cv+UDVOEYWSJ3Ya+FZu1gJwnYguVsU+P91YTAWJwFNopdEyB  
IVLGYkAXGjYwzaQeOObS82UUqWb7qbUMVWNhsA3Eedjn5n3e13vxoVQYpLSCcKQ  
ZH1KPHEAYj10ABcdSksS5hZr2yFsz/rFItav1GGund3AgMBAAGjIzAhMA4GA1Ud  
DwEB/wQEAwICBDAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBwUAA4IBAQAk  
eIau5j7D98m3Ltq9TIObWNZxY3ews39rYDRX3bMasxI9e0Af97N50Dzedd3YVD5  
Iye1tcolXz3NWBzBh3cxCrWMRJo05XglBwlZf1jbZwOpM9198z9VYz1Bv5GcEbeM  
wBmHcqyBFyYnrmRLW3j+tr9rb18Q2cJK/sw6I5jGbKsTJx44BkXia2XiE4e+OVWNg  
/OwNkxrfJS0beVbAN7qs80ByC/pcD233mq00/xya83D9yhnFDLr8nuZmImbX3xzsz  
szk/OhR8CRqYyJcIfYVXotZ1P84PdJA215wed2+dxWxz83KEBDSI5BeiW2d/tNi  
v0p9AyRqRMkYlW2a3az  
-----END CERTIFICATE-----
```

5 - Token

GitLab s'authentifie contre Kubernetes à l'aide de jetons de service, qui sont étendus à un `namespace` particulier.

Le jeton utilisé doit appartenir à un "service account" avec des privilèges `cluster-admin`.

Pour créer ce “service account”, créez un fichier appelé `gitlab-admin-service-account.yaml` avec le contenu suivant:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: gitlab-admin
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: gitlab-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: gitlab-admin
  namespace: kube-system
```

Appliquez le `ServiceAccount` et le `ClusterRoleBinding` à votre cluster:

```
> kubectl apply -f gitlab-admin-service-account.yaml
```

```
ahmedhosni_contact@cloudshell:~ (wuidigital-ci)$ kubectl apply -f gitlab-admin-service-account.yaml
serviceaccount/gitlab-admin unchanged
clusterrolebinding.rbac.authorization.k8s.io/gitlab-admin created
```

Récupérez le jeton pour le service account `gitlab-admin`:

```
> kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep gitlab-admin | awk '{print $1}')
```

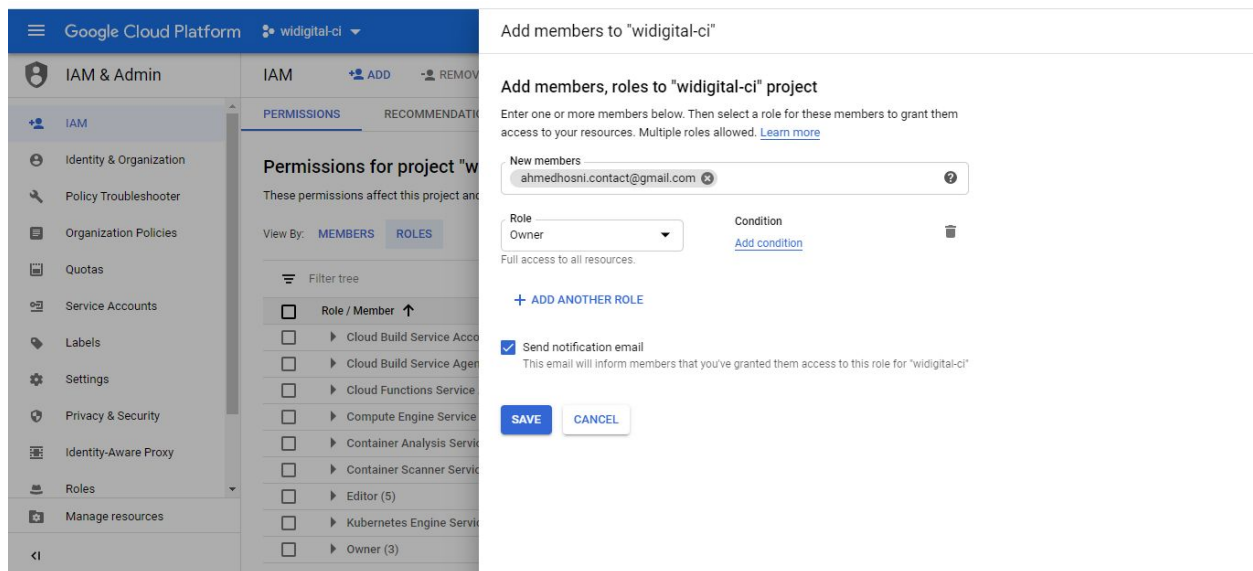
[illegible]

Copiez la valeur de la sortie `<authentication_token>`. **Assurez vous plus tard que le token est inséré sur une seule ligne et qu'il n'ya pas de caractères invisible, retours chariots, etc...**

Pour les clusters GKE:

vous aurez besoin de l'autorisation `container.clusterRoleBindings.create` pour créer un ClusterRoleBinding. Vous pouvez suivre la documentation de **Google Cloud** pour accorder l'accès.

Pour ce lab, nous avons simplement ajouté un nouveau membre au projet "widigital-ci" avec le role "Owner":



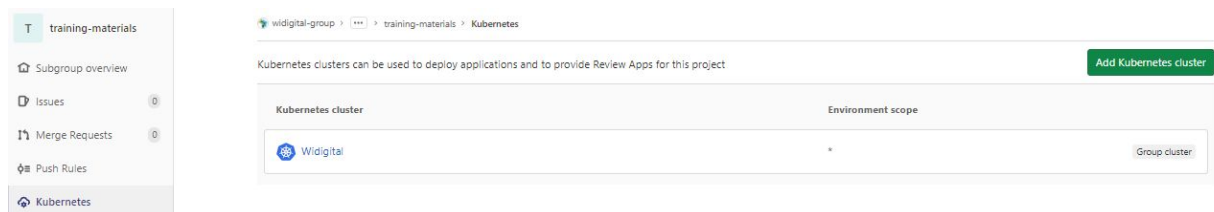
6- Cluster géré par GitLab

Laissez cette case cochée si vous souhaitez que GitLab gère les "namespace" et les "service account" pour ce cluster. Consultez la [section Clusters gérés](#) pour plus d'informations.

7- Namespace du projet (facultatif):

Vous n'avez pas à le remplir; en le laissant vide, GitLab en créera un pour vous.

8- Enfin, cliquez sur le bouton **Créer un cluster Kubernetes:**



Le contrôleur Ingress installé est [Ingress-NGINX](#), qui est pris en charge par la communauté Kubernetes.

Group cluster
Widigital

Details
Health
Applications
Advanced Settings

Choose which applications to install on your Kubernetes cluster. [More information](#)

Ingress
Ingress gives you a way to route requests to services based on the request host or path, centralizing a number of services into a single endpoint.

Uninstall

Web Application Firewall
Real-time web application monitoring, logging and access control. [More information](#)

☒

Global default
Set the global mode for the WAF in this cluster. This can be overridden at the environmental level.

Logging mode ▾

Ingress Endpoint

35.193.124.141

Point a wildcard DNS to this generated endpoint in order to access your application after it has been deployed. [More information](#)

• Installation de Gitlab Runner

Le runner Gitlab permet à vos jobs CI/CD d'être exécutés dans le cluster k8s. L'installation du runner avec Gitlab est simple, cliquez simplement sur le bouton Installer.

GitLab Runner
GitLab Runner connects to the repository and executes CI/CD jobs, pushing results back and deploying applications to production.

Uninstall

Crossplane
Crossplane enables declarative provisioning of managed services from your cloud of choice using `kubectl` or [GitLab Integration](#). Crossplane runs inside your Kubernetes cluster and supports secure connectivity and secrets management

Install

Une fois terminé, regardons à nouveau les pods du namespace **"gitlab-managed-apps"** et vous devriez voir un nouveau pod pour le runner Gitlab:

```

ahmedhosni_contact@cloudshell:~ (widigital-ci)$ kubectl get ns
NAME                                STATUS   AGE
cert-manager                        Active   36d
default                             Active   36d
gitlab-managed-apps                 Active   138m
kube-node-lease                     Active   36d
kube-public                         Active   36d
kube-system                         Active   36d
widigital-node-sample-20692662-production Active   69m
ahmedhosni_contact@cloudshell:~ (widigital-ci)$ kubectl get pods -n gitlab-managed-apps
NAME                                READY   STATUS    RESTARTS   AGE
ingress-nginx-ingress-controller-5956f57495-4qh6b  2/2     Running   0           138m
ingress-nginx-ingress-default-backend-5ddfb59666-vgfg7  1/1     Running   0           138m
prometheus-kube-state-metrics-76c5db74df-hk8l5        1/1     Running   0           137m
prometheus-prometheus-server-d8c55b754-k2c67         2/2     Running   0           137m
runner-gitlab-runner-9f46bfd86-56hhl                 1/1     Running   0           138m

```


Vous pouvez vérifier que le runner est connecté à votre projet en consultant la section Paramètres → CI/CD → Runners dans Gitlab:

The screenshot shows the GitLab interface for the 'training-materials' project. The left sidebar contains navigation links: Subgroup overview, Issues, Merge Requests, Push Rules, Kubernetes, Packages & Registries, Analytics, Members, Settings (selected), General, Projects, Repository, CI / CD (selected), Webhooks, Audit Events, and Billing. The main content area is titled 'Set up a group Runner automatically' and 'Set up a group Runner manually'. Below these, it shows 'Available Runners: 1' with a table of runners.

Type/State	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact
group	1UBQzdAI		13.2.2	35.238.187.140	n/a	0	cluster, kubernetes	21 minutes ago

The screenshot shows the details for Runner #2717585. The breadcrumb trail is 'widigital-group > training-materials > Runners'. The runner is identified as 'Group'. The properties and values are as follows:

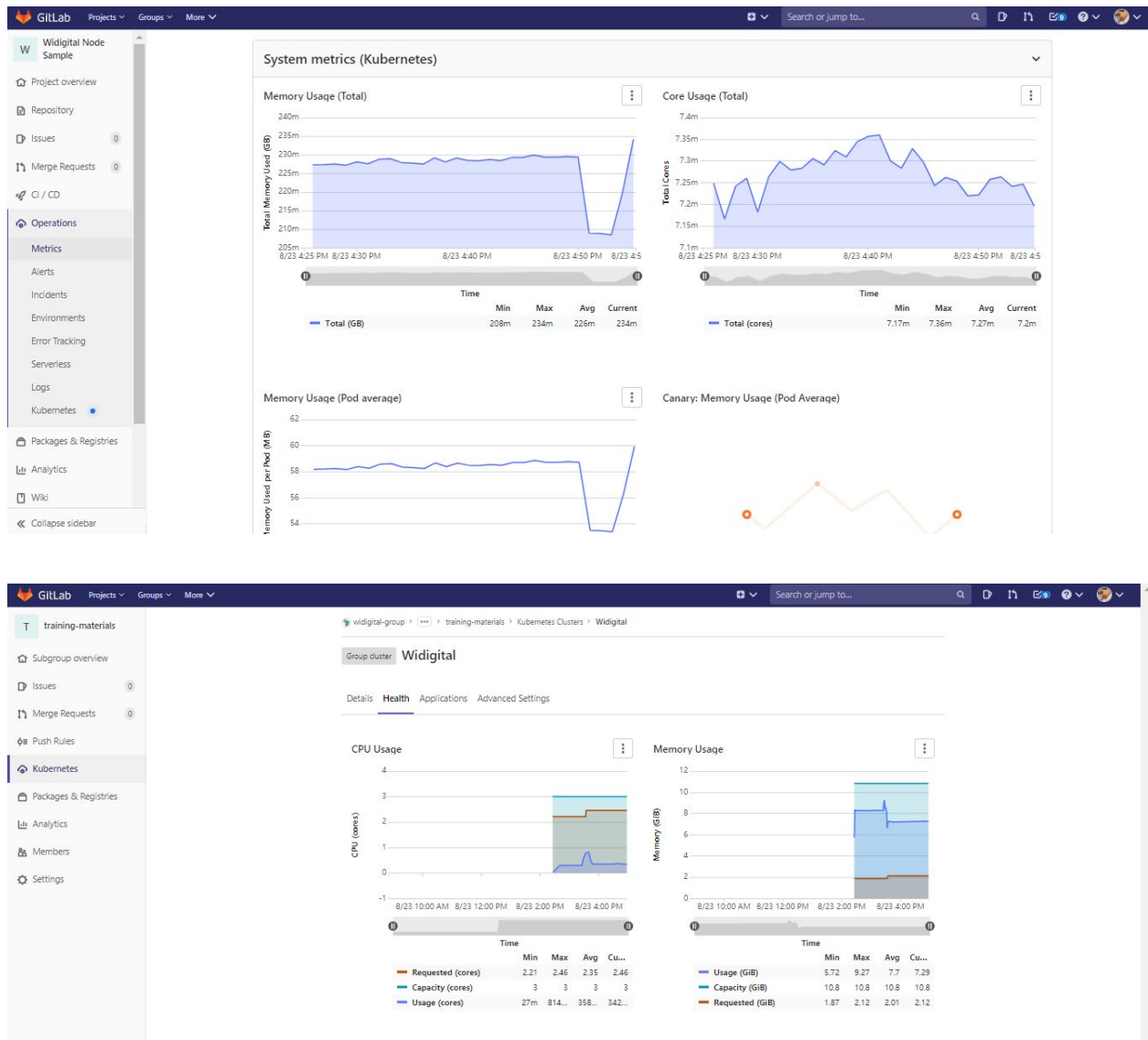
Property Name	Value
Active	Yes
Protected	No
Can run untagged jobs	Yes
Tags	cluster, kubernetes
Name	kubernetes-cluster
Version	13.2.2
IP Address	35.238.187.140
Revision	a998cadd
Platform	linux
Architecture	amd64
Description	
Maximum job timeout	
Last contact	23 minutes ago

● Installation de Prometheus

Prometheus est un système de surveillance et d'alerte open-source utile pour superviser vos applications déployées.

GitLab est capable de surveiller automatiquement les applications, en utilisant l'intégration Prometheus .

Les métriques de CPU et de mémoire du conteneur Kubernetes sont automatiquement collectées, et les métriques de réponse sont également récupérées à partir de NGINX Ingress.



Remarque: Le Helm chart stable/prometheus est utilisé pour installer cette application avec un fichier values.yaml

- **Installation de cert-manager**

[cert-manager](#) est un contrôleur de gestion de certificats Kubernetes natif qui facilite l'émission de certificats. L'installation de cert-manager sur votre cluster émettra un certificat par [Let's Encrypt](#) et garantira que les certificats sont valides et à jour.

- cert-manager est installé à l'aide de GitLab CI / CD en définissant la configuration dans `.gitlab/managed-apps/config.yaml`.
- Est installé dans l' `gitlab-managed-apps` namespace de votre cluster.
- Peut être installé avec ou sans un `ClusterIssuer` par défaut de `Let's Encrypt`, qui nécessite la spécification d'une adresse e-mail. L'adresse e-mail est utilisée par Let's Encrypt pour vous contacter à propos de l'expiration des certificats et des problèmes liés à votre compte.

10- Créer un projet Node Express à partir d'un Template:

The screenshot shows the GitLab 'New project' interface. On the left, there's a sidebar with 'New project' and some instructions. The main area has four tabs: 'Blank project', 'Create from template' (selected), 'Import project', and 'CI/CD for external repo'. Under 'Create from template', there's a search bar with 'NodeJS Express' selected. Below that, the 'Project name' is 'Widigital Node Sample'. The 'Project URL' is 'https://gitlab.com/widigital-group/administration/training-materials/widigital-node-sample' and the 'Project slug' is 'widigital-node-sample'. There's a 'Project description (optional)' field and a 'Visibility Level' dropdown set to 'Private'. At the bottom, there are 'Create project' and 'Cancel' buttons.

<https://gitlab.com/widigital-group/administration/training-materials/widigital-node-sample>

11- Activer Auto Devops:

https://gitlab.com/widigital-group/administration/training-materials/widigital-node-sample/-/settings/ci_cd#autodevops-settings

Alors que "Auto DevOps" est activé par défaut, Auto DevOps peut être désactivé au niveau du groupe. Suivez ces étapes pour activer Auto DevOps s'il est désactivé:

- Accédez à **Paramètres > CI / CD > DevOps automatique**, puis cliquez sur **Développer**
- Sélectionnez **Par défaut au pipeline DevOps automatique** pour afficher plus d'options.
- Dans **Stratégie de déploiement**, sélectionnez la stratégie de déploiement continue souhaitée pour déployer l'application en production une fois le pipeline exécuté avec succès sur la branche master.

- Cliquez sur **Enregistrer les modifications**.

Auto DevOps

Auto DevOps can automatically build, test, and deploy applications based on predefined continuous integration and delivery configuration. [Learn more about Auto DevOps](#) or use our [quick start guide](#) to get started right away.

Collapse

☒ Default to Auto DevOps pipeline

The Auto DevOps pipeline will run if no alternative CI configuration file is found. [More information](#)

Deployment strategy

- ☒ Continuous deployment to production
- ☐ Continuous deployment to production using timed incremental rollout
- ☐ Automatic deployment to staging, manual deployment to production

Save changes

Un build se déclenche immédiatement dès que vous sauvegardez vos changement:

The screenshot shows the GitLab Pipelines interface for the 'widigital-group' project. The breadcrumb trail is 'widigital-group > training-materials > Widigital Node Sample > Pipelines'. At the top, there are tabs for 'All' (selected), 'Finished', 'Branches', and 'Tags'. To the right are buttons for 'Run Pipeline', 'Clear Runner Caches', and 'CI Lint'. Below these is a search bar labeled 'Filter pipelines'. The main table lists pipeline runs with columns for Status, Pipeline, Triggerer, Commit, and Stages. One pipeline is shown with status 'running', ID '#181015310', triggerer 'latest Auto DevOps', and commit 'master cca5d1d2'. The stages listed are 'Initial template creation'. A red 'X' button is visible on the right side of the table.

Ce build contient le pipeline suivant:

The screenshot shows the GitLab CI/CD pipeline configuration page for the 'Widigital Node Sample' project. The left sidebar contains navigation links: Project overview, Repository, Issues (0), Merge Requests (0), CI / CD (selected), Pipelines, Jobs, Schedules, Operations, Packages & Registries, Analytics, Wiki, Snippets, Members, and Settings. The main content area shows the pipeline configuration for 'Initial template creation'. It indicates '9 jobs for master' and lists the jobs: 'code_quality', 'eslint-sast', 'nodejs-scan-sast', 'secret_detection_', 'secrets-sast', and 'test'. The pipeline is currently 'running' and was triggered 1 minute ago by 'Ahmed HOSNI'. The pipeline configuration is shown as a DAG (Directed Acyclic Graph) with stages: Test, Production, and Performance. The 'Test' stage includes jobs like 'code_quality', 'eslint-sast', 'nodejs-scan-sast', 'secret_detection_', 'secrets-sast', and 'test'. The 'Production' stage includes 'production'. The 'Performance' stage includes 'performance'. The pipeline is currently running on the 'master' branch with commit 'cca5d1d2'.

Ce pipeline regroupe 9 jobs:

Pipeline	DAG	Jobs 9	Tests 0
Status	Job ID	Name	Coverage
Build			
passed	#699375206	build	00:01:02 2 minutes ago
Test			
running	#699375208 allowed to fail	code_quality	00:03:30
passed	#699375209	eslint-sast	00:00:20 2 minutes ago
passed	#699375210	nodejs-scan-sast	00:00:17 2 minutes ago
passed	#699375212	secret_detection_default_branch	00:00:19 2 minutes ago
passed	#699375211	secrets-sast	00:00:23 2 minutes ago
passed	#699375207	test	00:01:34 1 minute ago
Production			
created	#699375213	production	

Vous pouvez aussi vérifier le registre des conteneurs pour voir les images créées durant l'étape de build:

https://gitlab.com/widigital-group/administration/training-materials/widigital-node-sample/container_registry

The screenshot shows the GitLab interface for the Container Registry. The left sidebar contains navigation links: Project overview, Repository, Issues, Merge Requests, CI / CD, Operations, Packages & Registries (selected), Analytics, Wiki, Snippets, Members, and Settings. The main content area displays the path: widigital-g... > administr... > training-mat... > Widigital Node Sa... > Container Re... > widigital-group/administration/training-materials/widigital-node-sample/master. Below this, the title is "widigital-group/administration/training-materials/widigital-node-sample/master tags". There are two image tags listed: "cca5d1d2fb24ad6e945aa56f2c848c76a2cc1c61" (29.64 MiB) and "latest" (29.64 MiB). Both tags were published 5 minutes ago with a digest of a8f55bf. A "Delete selected" button is visible in the top right corner of the tags list.

12- Déploiement en production:

The screenshot shows the GitLab CI/CD interface for a job named "production". The job is currently running, triggered by Ahmed HOSNI. The job description states: "This job is creating a deployment to production using cluster WIdigital and namespace wIdigital-node-sample-20692662-production." The job logs show the following steps:

1. Running with gitlab-runner 13.3.0-rc1 (669fc507)
2. on docker-auto-scale z3MlRhu-
3. Preparing the "docker-machine" executor
4. Using Docker executor with image registry.gitlab.com/gitlab-org/cluster-integration/auto-deploy-image:v1.0.0 ...

The right sidebar shows the job details for "production":

- Duration: 22 seconds
- Timeout: 1h (from project)
- Runner: shared-runners-manager-7 (#2568000)
- Job artifacts: These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.
- Download and Browse buttons
- Commit cca5d1d2
- Initial template creation
- Pipeline #181015310 for master
- production (dropdown)
- production

The screenshot shows the GitLab CI/CD interface for a job named "production". The job is completed, showing the following logs:

```
101 Deploying new stable release...
102 Release "production" does not exist. Installing it now.
103 NAME: production
104 LAST DEPLOYED: Sun Aug 23 14:33:29 2020
105 NAMESPACE: wIdigital-node-sample-20692662-production
106 STATUS: DEPLOYED
107 RESOURCES:
108 ==> v1/Deployment
109 NAME READY UP-TO-DATE AVAILABLE AGE
110 production 1/1 1 1 18s
111 ==> v1/Pod(related)
112 NAME READY STATUS RESTARTS AGE
113 production-79d7f9f7c6-2jxnl 1/1 Running 0 18s
114 ==> v1/Service
115 NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
116 production-auto-deploy ClusterIP 10.7.245.123 <none> 5000/TCP 18s
117 ==> v1beta1/Ingress
118 NAME HOSTS ADDRESS PORTS
119 AGE
120 production-auto-deploy wIdigital-group-administration-training-materials-wIdigital-nod.23.236.53.242.nip.io,le-20692662.23.236.53.242.nip.io 34.68.80.96 80, 44
121 18s
122 NOTES:
123 Application should be accessible at
124 http://wIdigital-group-administration-training-materials-wIdigital-nod.23.236.53.242.nip.io
125 deployment "production" successfully rolled out
126 $ auto-deploy delete canary
127 $ auto-deploy delete rollout
128 $ auto-deploy persist_environment_url
129 Uploading artifacts for successful job
130 Uploading artifacts...
131 environment_url.txt: found 1 matching files and directories
132 WARNING: tiller.log: no matching files
133 Uploading artifacts as "archive" to coordinator... ok id=699375213 responseStatus=201 Created token=RkswZ7uR
134 Job succeeded
```

The right sidebar shows the job details for "production":

- Duration: 1 minute 51 seconds
- Timeout: 1h (from project)
- Runner: shared-runners-manager-7 (#2568000)
- Job artifacts: These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.
- Keep, Download, and Browse buttons
- Commit cca5d1d2
- Initial template creation
- Pipeline #181015310 for master
- production (dropdown)
- production

13- Surveillez votre projet

Après avoir déployé avec succès votre application, vous pouvez afficher son site Web et vérifier son état de santé sur la page **Environnements** en accédant à **Opérations> Environnements**.

Cette page affiche des détails sur les applications déployées et la colonne de droite affiche des icônes qui vous relient aux tâches d'environnement courantes:

widigital-group > ... > training-materials > Wigital Node Sample > Environments

Available 1 Stopped 0 New environment

Environment	Deployment	Job	Commit	Updated	Auto stop in
production	#1 by	production #699375213	Y master -> cca5d1d2 Initial template creation	2 hours ago	

Upgrade plan to unlock Canary Deployments feature

Canary Deployments is a popular CI strategy, where a small portion of the fleet is updated to the new version of your application. [Read more](#)

[Contact sales to upgrade](#)

- **Ouvrir l'environnement en direct:** Ouvre l'URL de l'application déployée en production
- **Monitoring:** Ouvre la page de métriques où Prometheus collecte des données sur le cluster Kubernetes et comment l'application l'affecte en termes d'utilisation de la mémoire, d'utilisation du processeur et de latence
- **Déployer vers:** Affiche une liste d'environnements dans lesquels vous pouvez effectuer un déploiement
- **Terminal:** Ouvre une session de terminal Web à l'intérieur du conteneur où l'application s'exécute
- **Redéployer dans l'environnement:** Pour plus d'informations, voir Réessayer et revenir en arrière
- **Arrêter l'environnement:** Pour plus d'informations, voir Arrêt d'un environnement