

app.yaml. Ensuite, dans le dossier où se trouve ce fichier de configuration, ~~on exécute la~~ commande suivante dans le terminal : gcloud app deploy.

Les modules Node.js sont installés dans le cloud tels qu'ils sont répertoriés dans le fichier package.json et notre service est lancé en utilisant npm start.

La figure 33 représente la configuration de notre service web.



```

runtime: nodejs10
env: standard
instance_class: F1
handlers:
- url:.*
  script: auto
env_variables:
  GCLOUD_STORAGE_BUCKET: stockagedefichiers
automatic_scaling:
  min_idle_instances: automatic
  max_idle_instances: automatic
  min_pending_latency: automatic
  max_pending_latency: automatic
  max_instances: 1
network: {}

```

**Figure 33 :** La configuration de notre service web

Comme indiqué dans la figure ci-dessus, nous avons choisi de déployer notre service web sur un environnement standard de produit App Engine. L'environnement standard App Engine est basé sur des instances de conteneur qui s'exécutent sur l'infrastructure de Google. L'environnement standard App Engine permet de concevoir et de déployer des applications qui s'exécutent de manière fiable, même lorsqu'elles doivent faire face à des charges importantes et gérer de grandes quantités de données.

Les applications s'exécutent dans un environnement de bac à sable sécurisé, ce qui permet à l'environnement standard App Engine de répartir les requêtes sur plusieurs serveurs et d'adapter ces derniers à la demande de trafic. Celui-ci est indépendant du matériel, du système d'exploitation et de l'emplacement physique du serveur.

Concernant les tarifs, un quota gratuit pour l'utilisation des ressources App Engine s'applique aux applications exécutées dans l'environnement standard. Au-delà de cette limite, l'utilisation des ressources App Engine est facturée comme décrit la figure suivante :