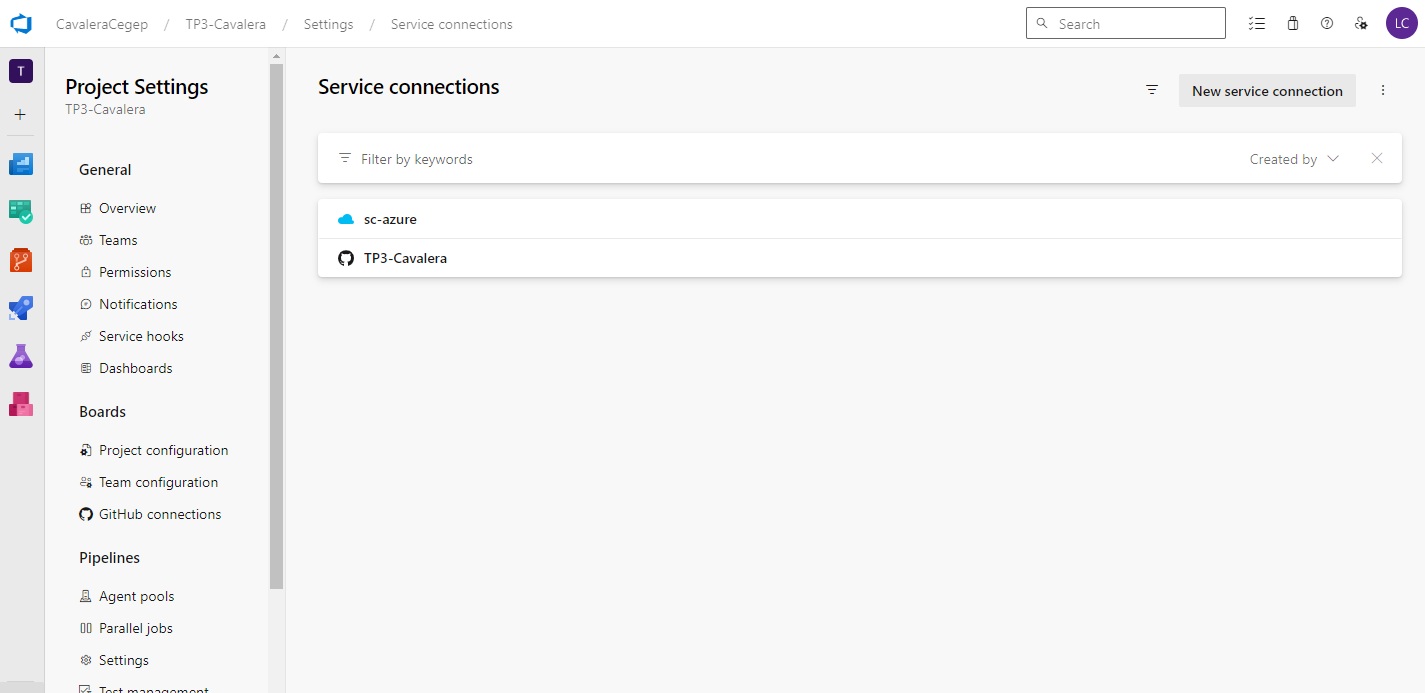
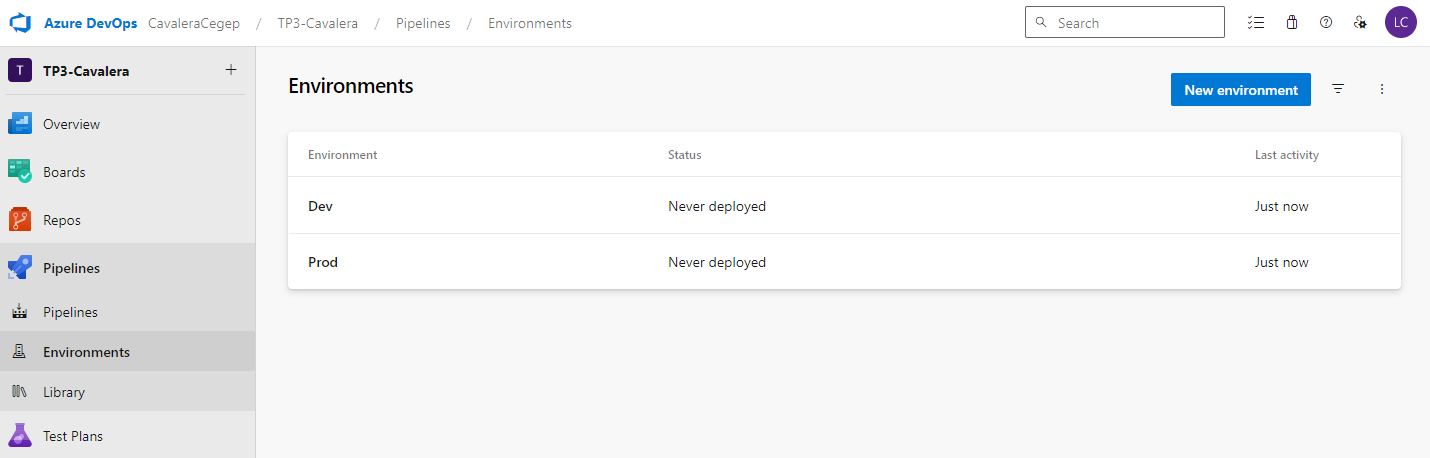
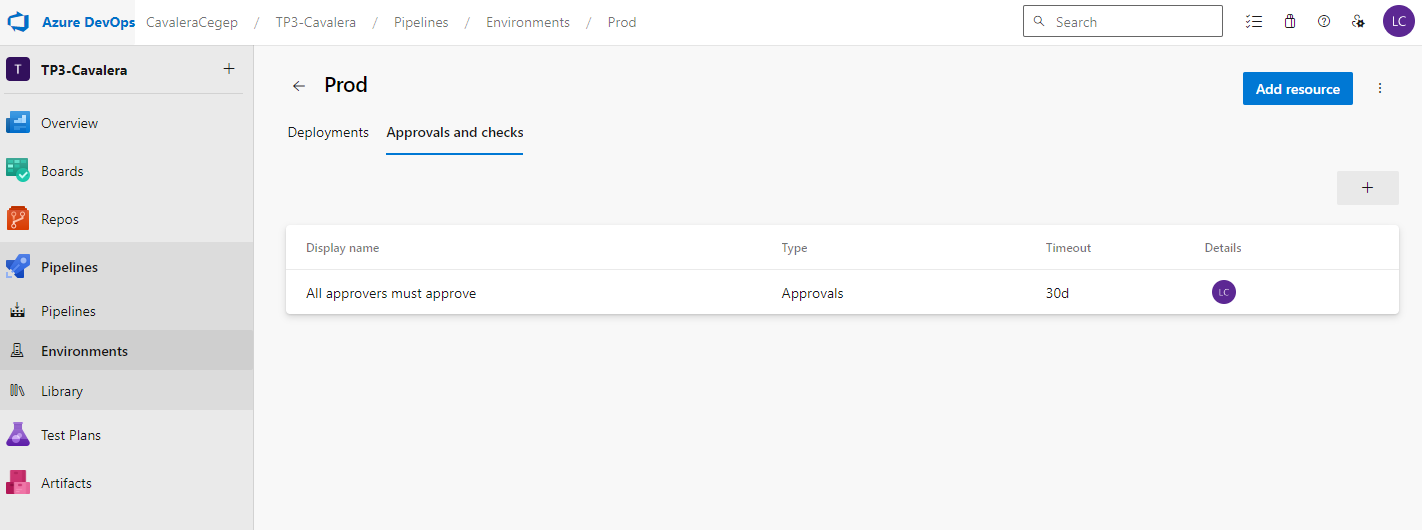
**Exercice1 :**

Ajout du service de connexion Azure (« sc-azure »).



Ajout des deux environnements Dev et Prod et ajout de l’approbation dans l’environnement Prod.





**Exercice2 :**

Pour répondre à ce besoin il faudrait utiliser le service d’intégration Logic Apps et un Event Grid pour les raisons suivantes :

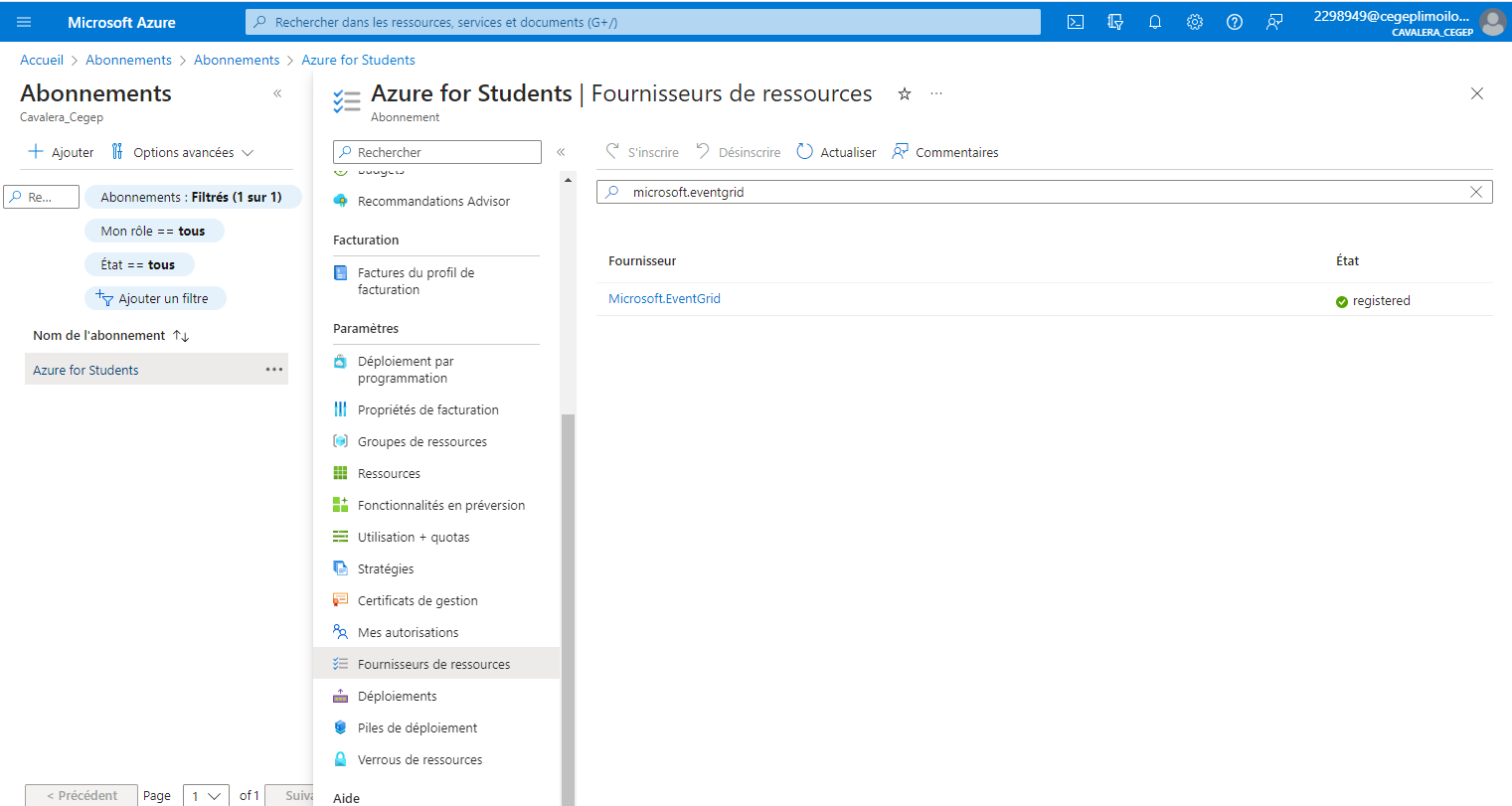
Pour la Logic App :

* La suppression d’un document è un évent qui n’arrive pas souvent donc ce n’est pas nécessaire provisionner l’entreprise de ressources supplémentaires. Donc la solution Logic Apps qu’elle est server-less répond à nos besoins.
* La facturation est à l’utilisation, donc on n’a pas besoin de dépenser sur de ressources qui ne seront pas utilisé souvent. De plus avec le plan Consommation on 1000000 d’exécution par mois gratuits.
* Service semple à implémenter pour déclencher l’envoie d’un courriel à la suite d’un événement dans le compte de stockage.

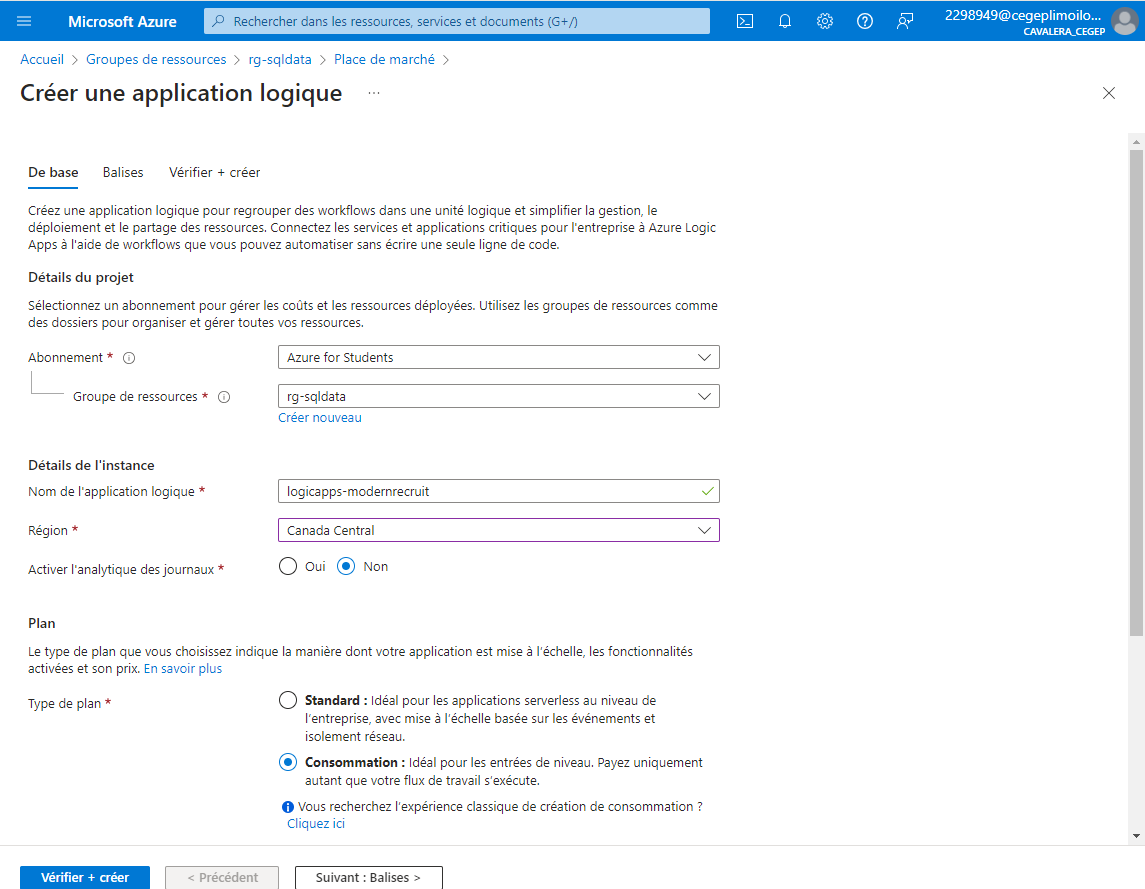
Pour Event Grid :

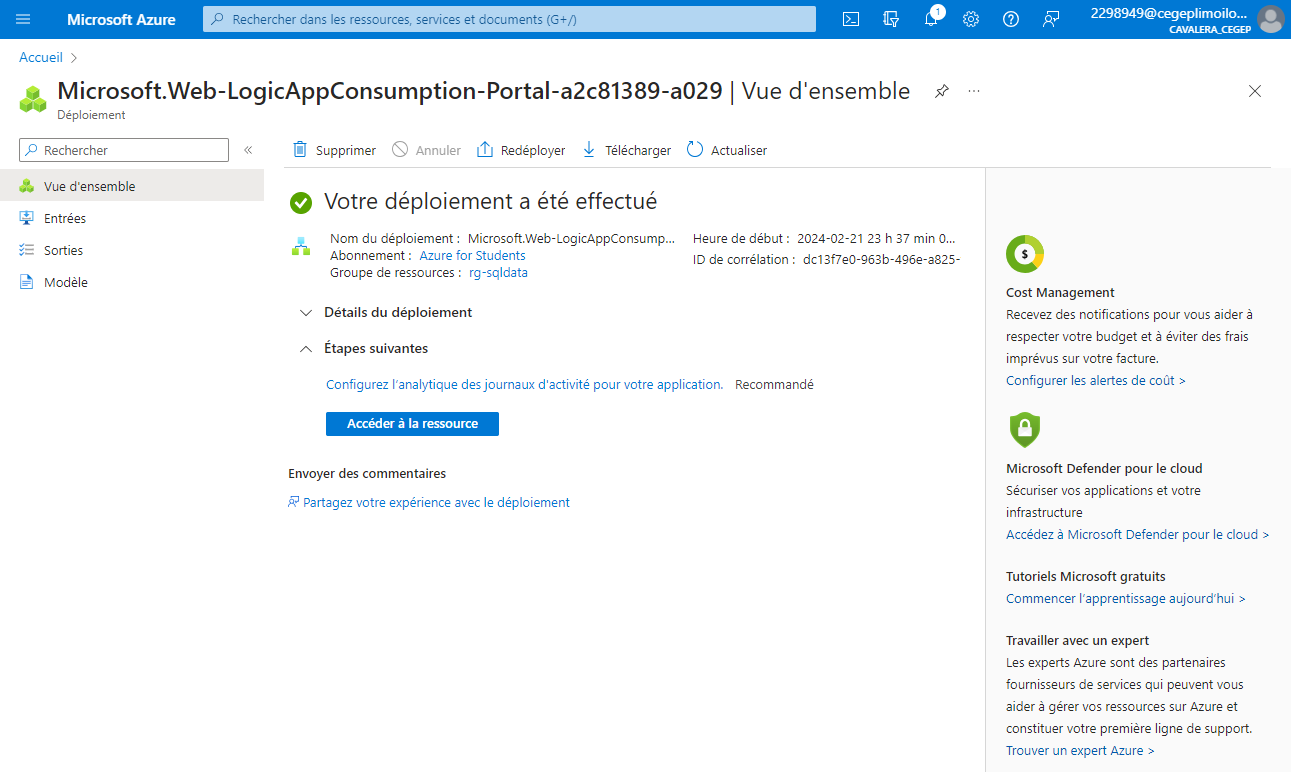
* N’existe pas un déclencheur pour la suppression de blob du compte de stockage donc il faut utiliser Event Grid
* Il est basé sur l’évènements et dispose d’une prise en charge intégrée pour les évènements provenant des services Azure
* Pas besoin de configurer notre propre déclencheur de la Logic App pour répondre à notre besoin spécifique

Activation de Azure Event Grid dans l’abbonement



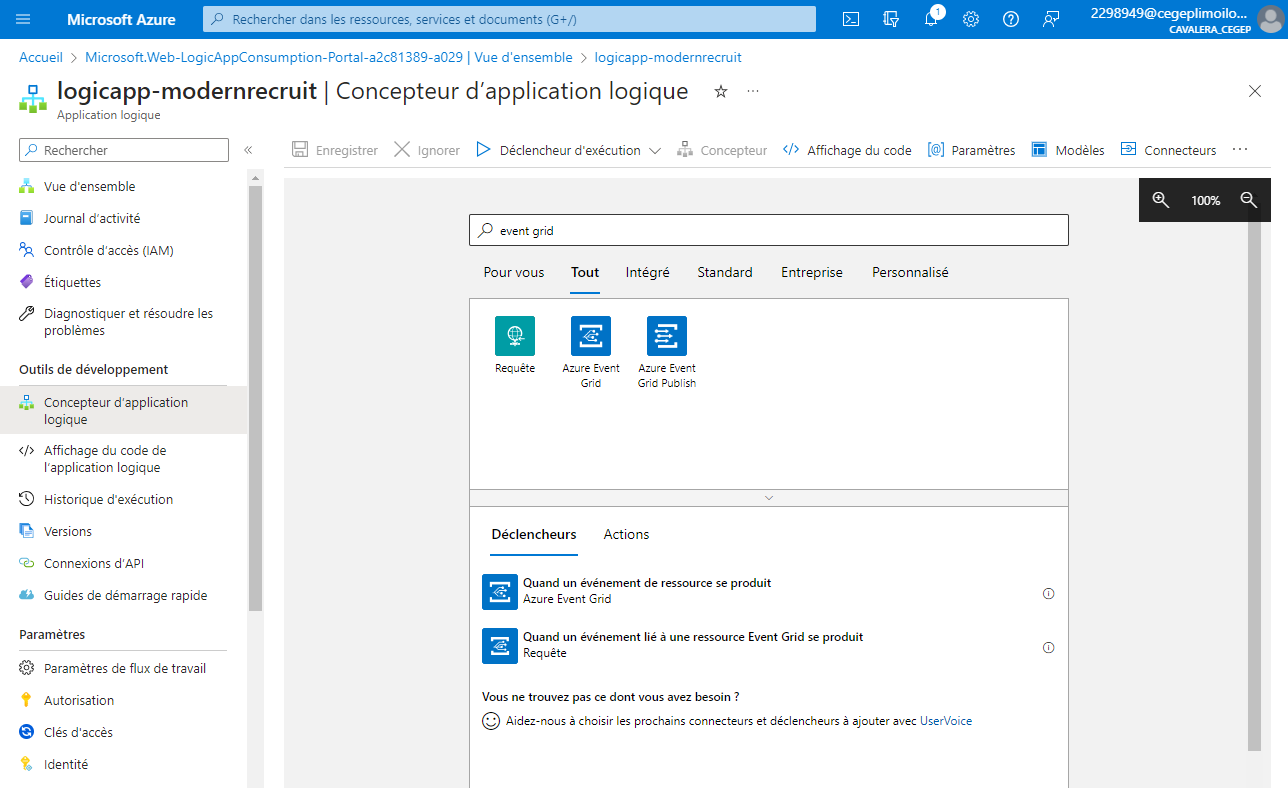
Création de la Logicapp avec le plan Consommation.



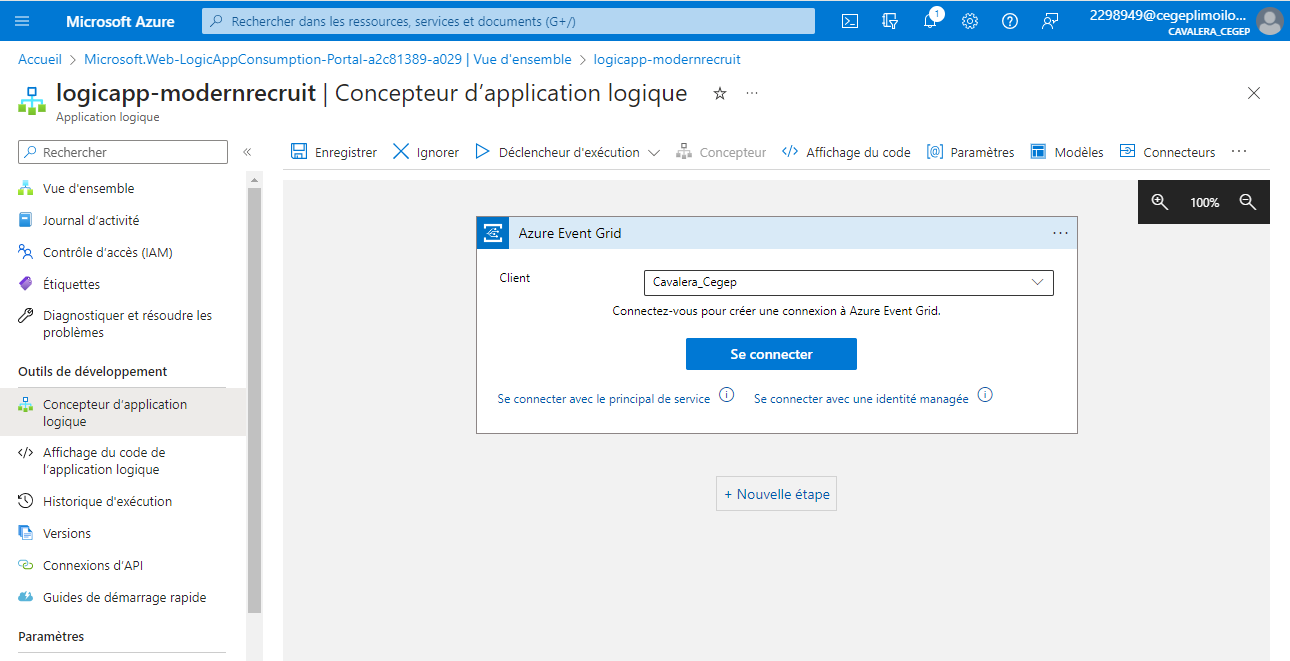


Création du workflow de l’application logique pour répondre à mon besoin.

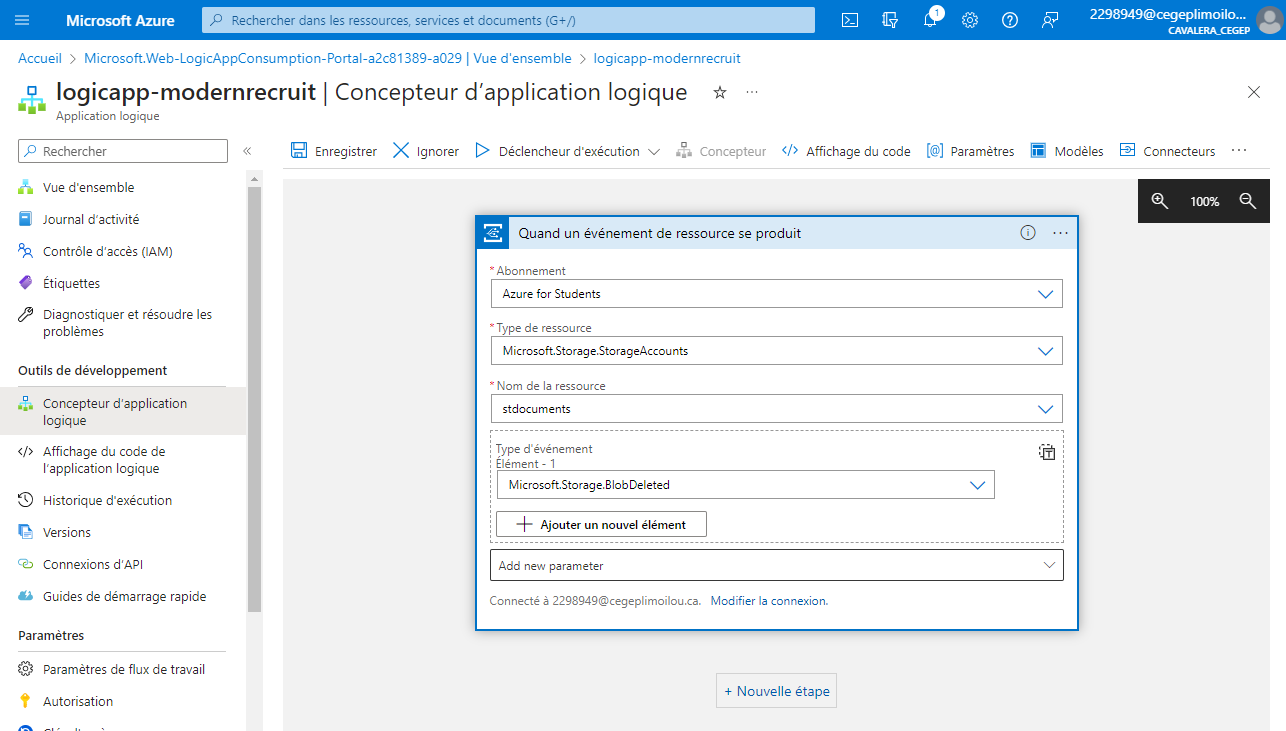
Utilisation du connecteur Azure Event Grid



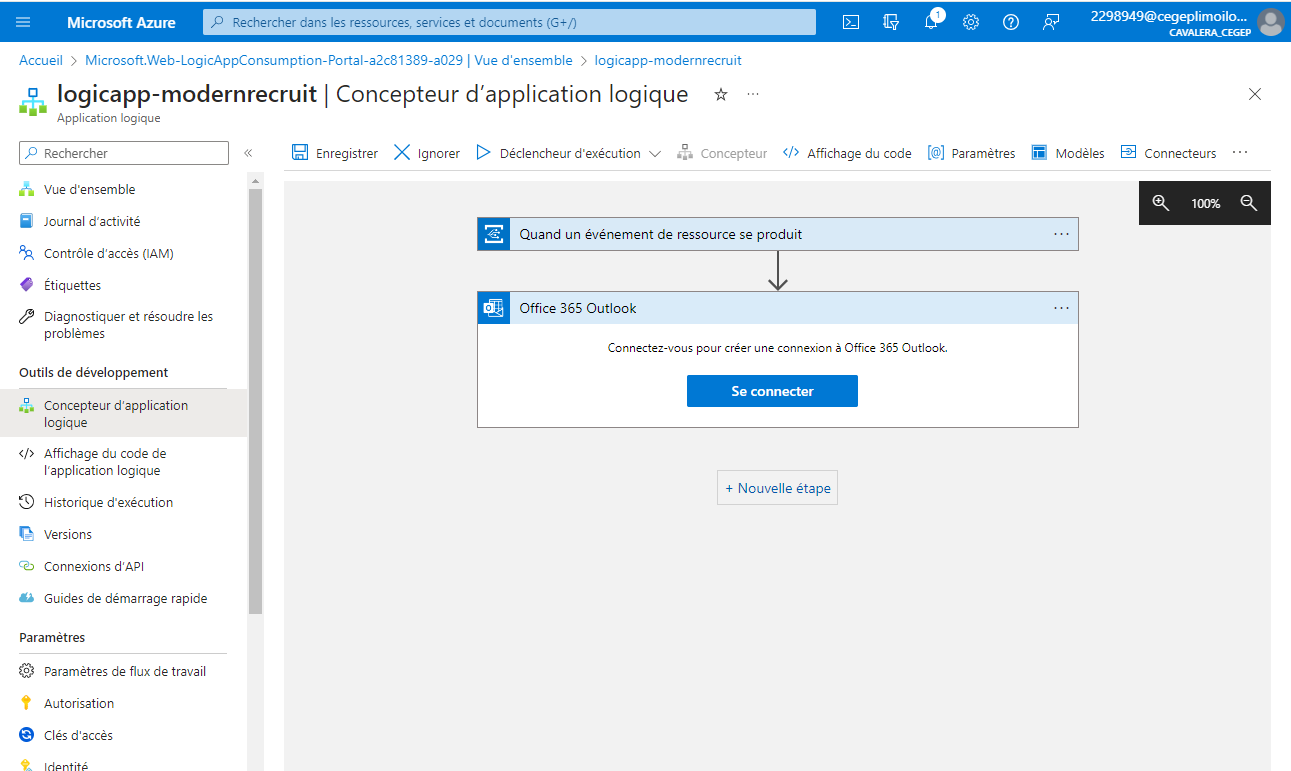
Se connecter au service Azure Event Grid



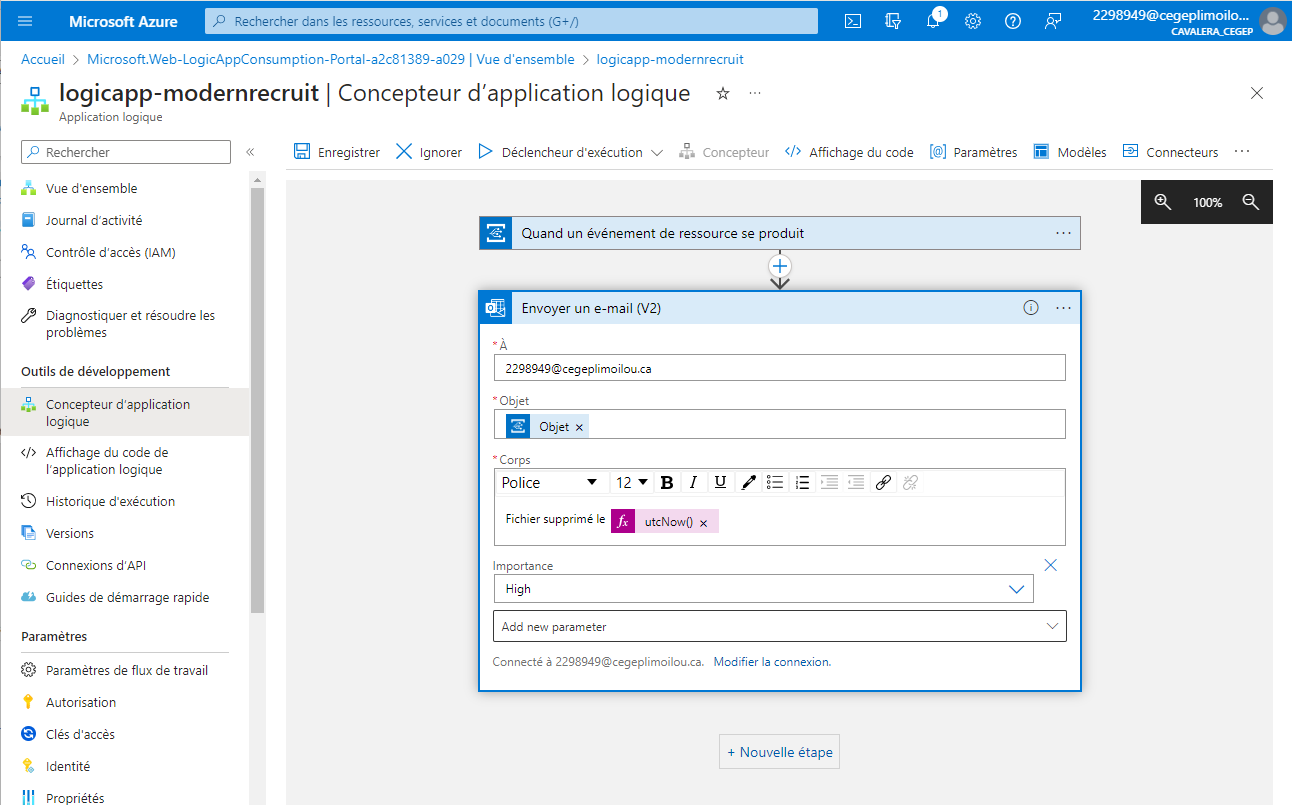
Configuration du déclencheur Azure Event Grid pour s’exécuter à la suppression d’un blob dans le compte de stockage Documents (stdocuments)



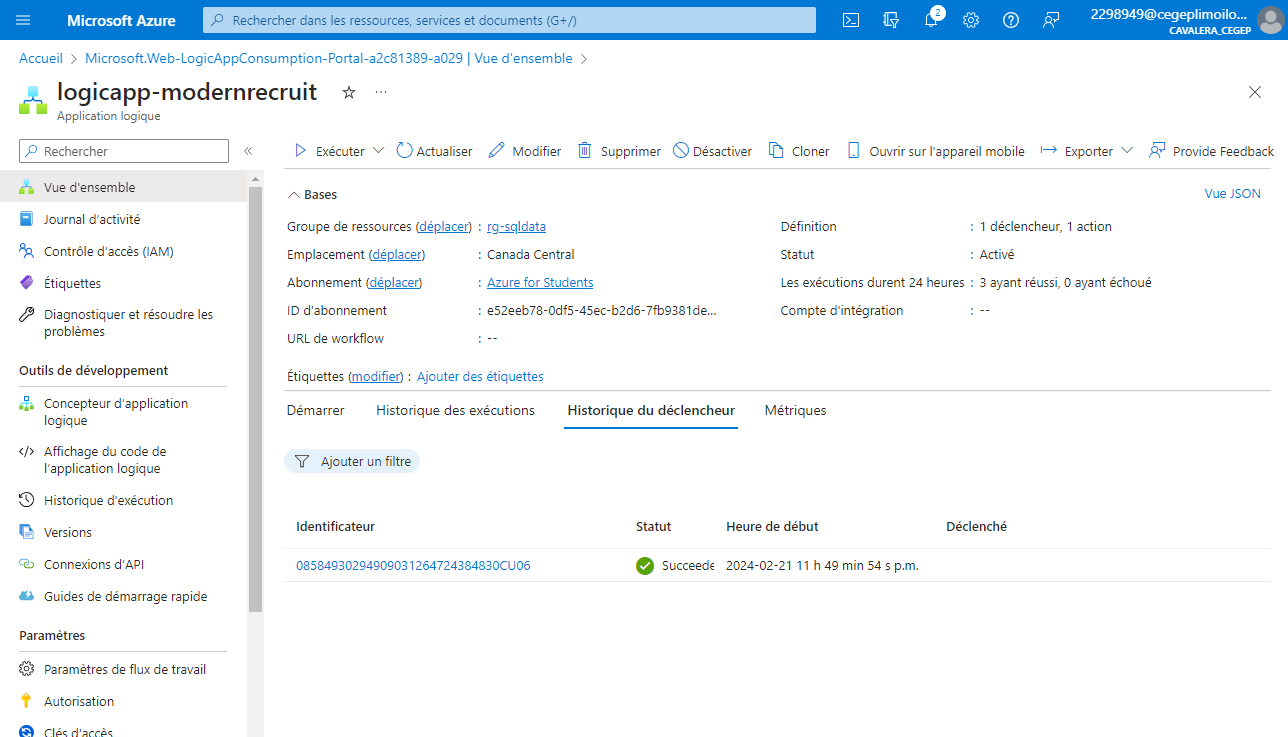
Se connecter au service Office 365 Outlook pour l’envoi de courriels.



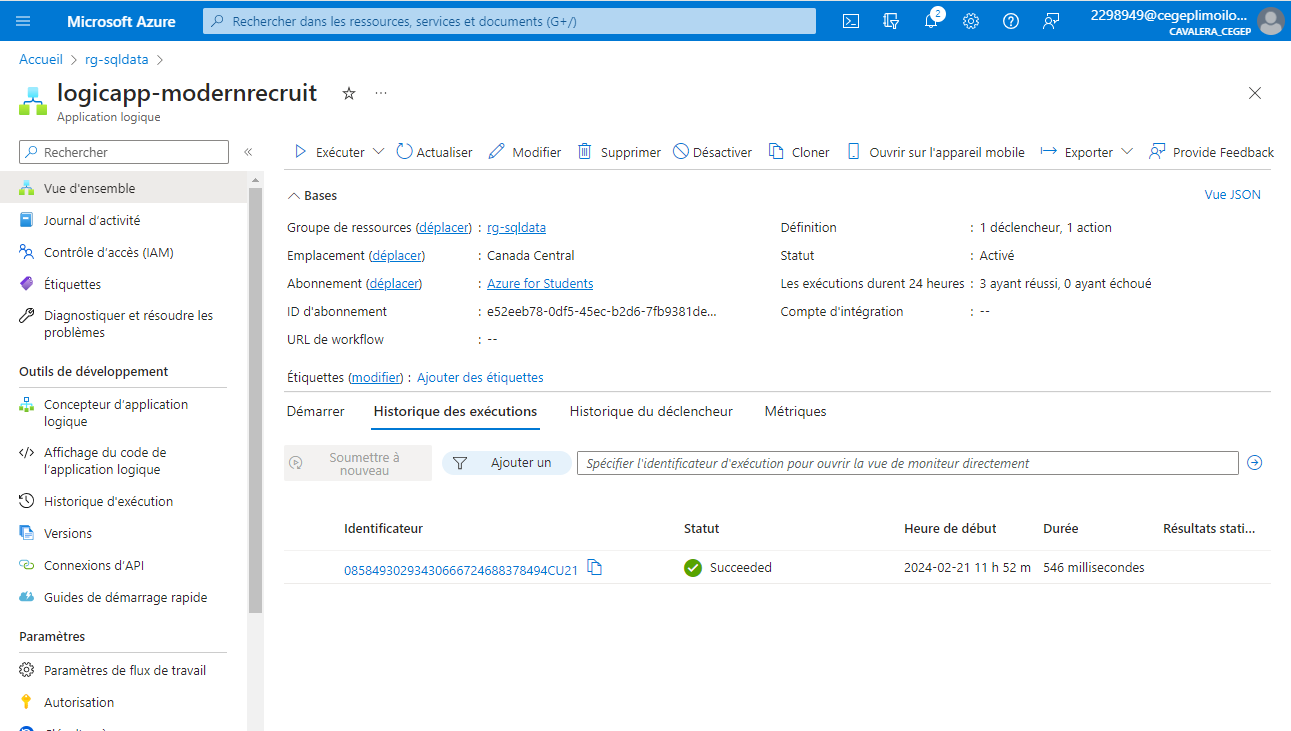
Configuration de l’action pour envoyer le courriel.



Enregistrement de la Logic App

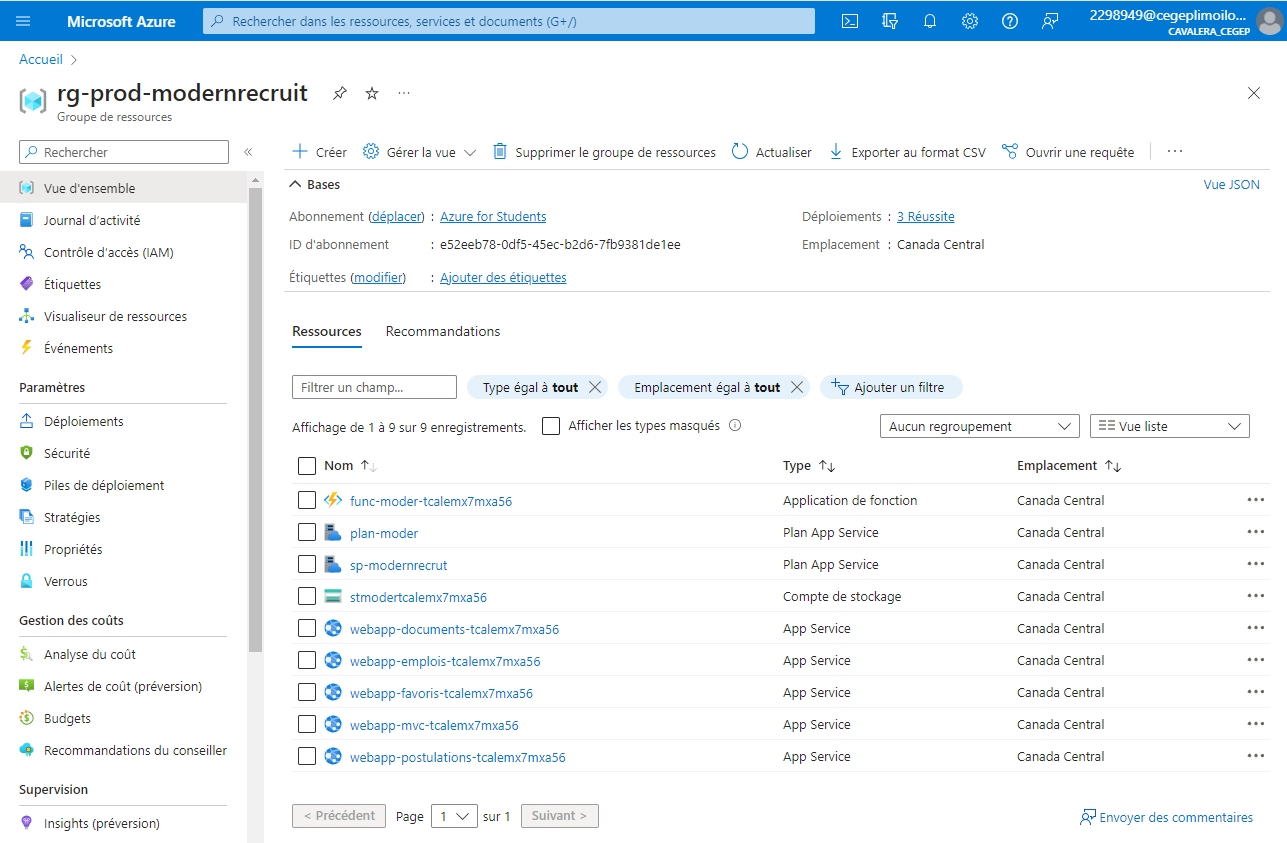


Exécution avec succès de l’envoi du courriel lorsqu’un document est supprimé du compte de stockage.

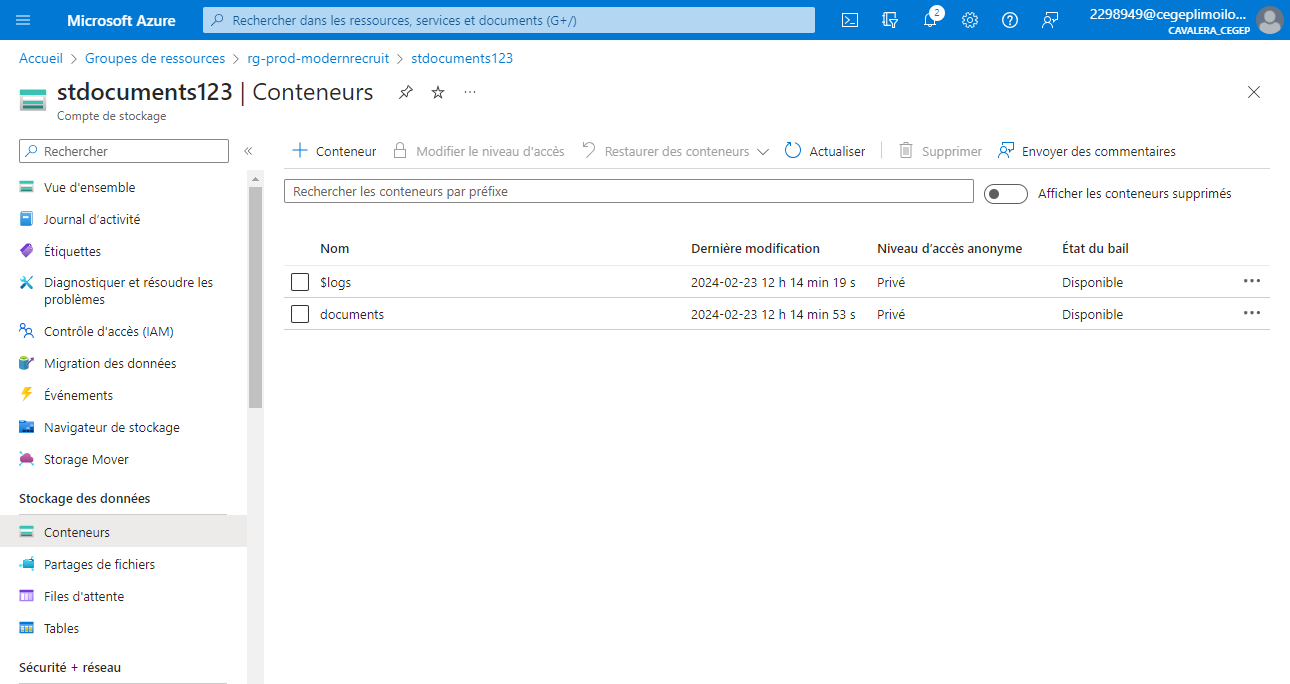


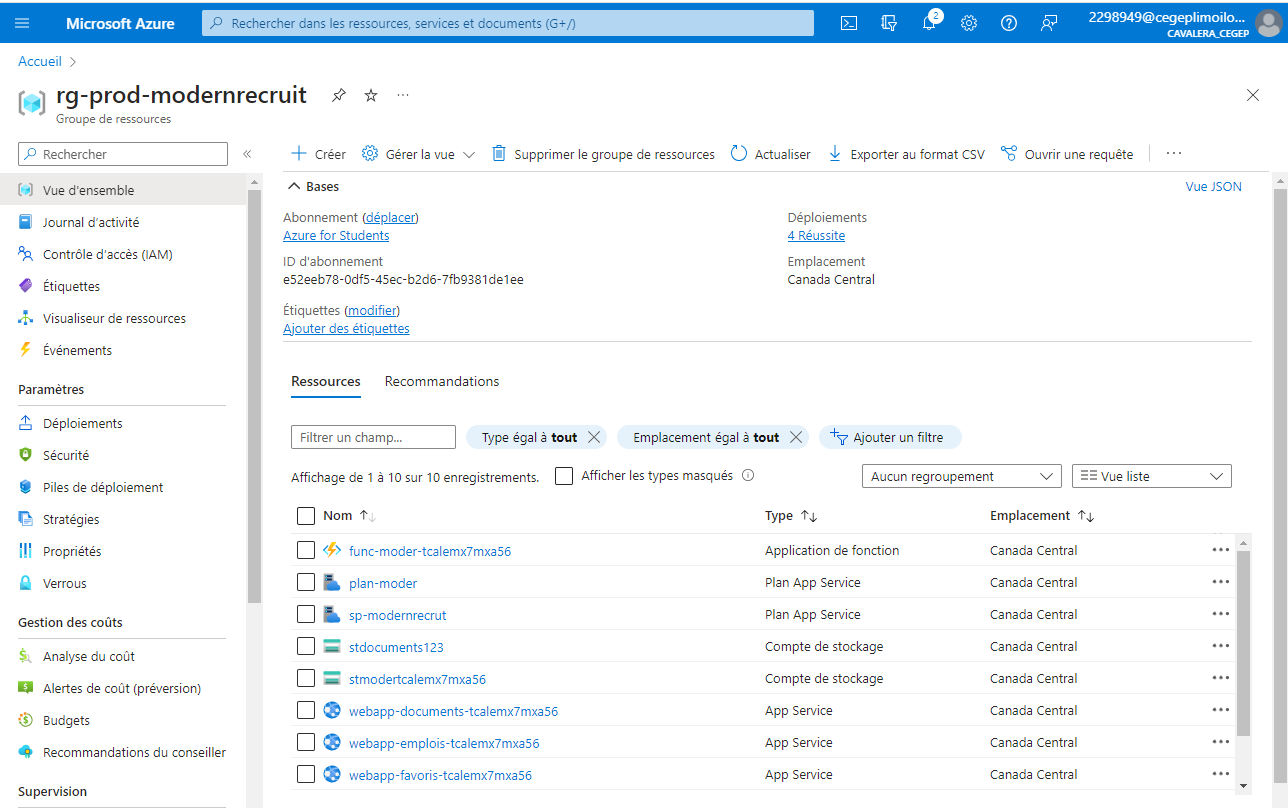
**Exercice3 :**

Exécution du script Bicep pour créer l’application de fonction dans le groupe de ressources Modern Recruit.

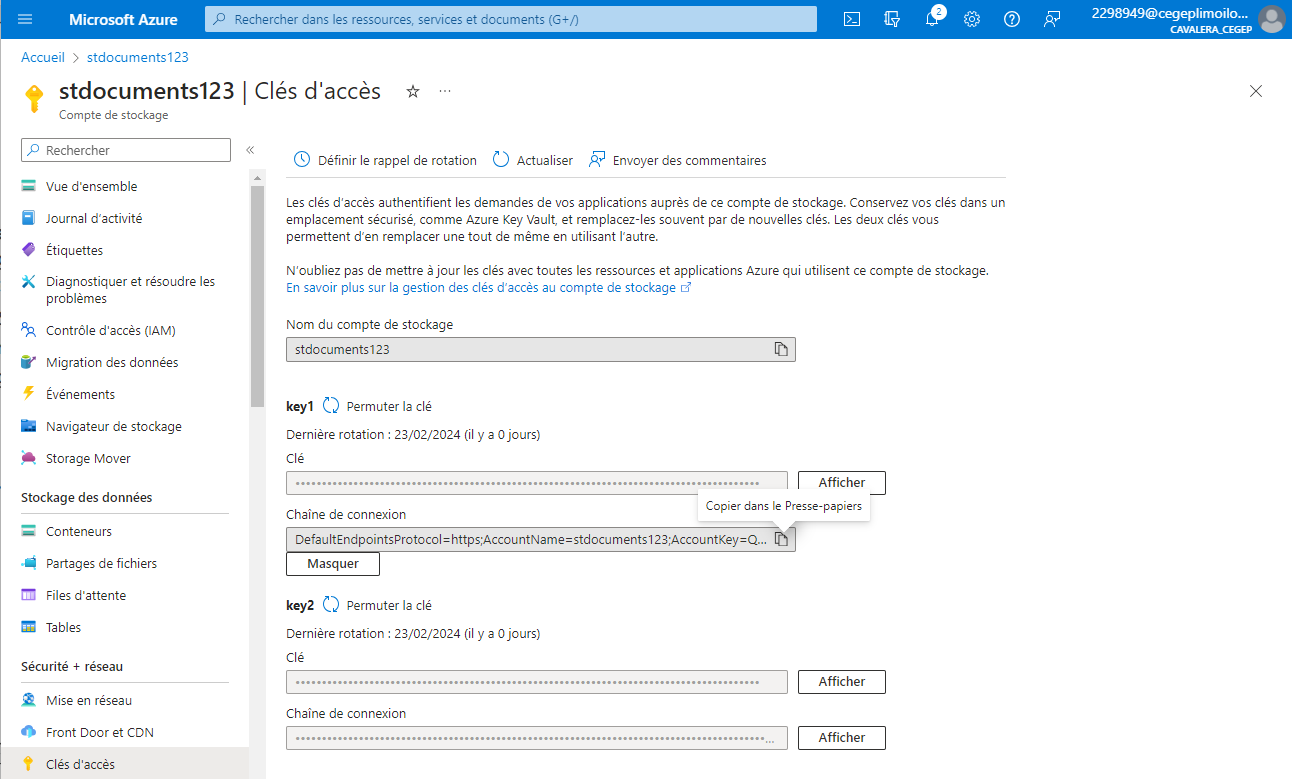


Création du compte de stockage dans le groupe de ressources Modern Recruit avec un conteneur pour enregistrer les documents par la app function.

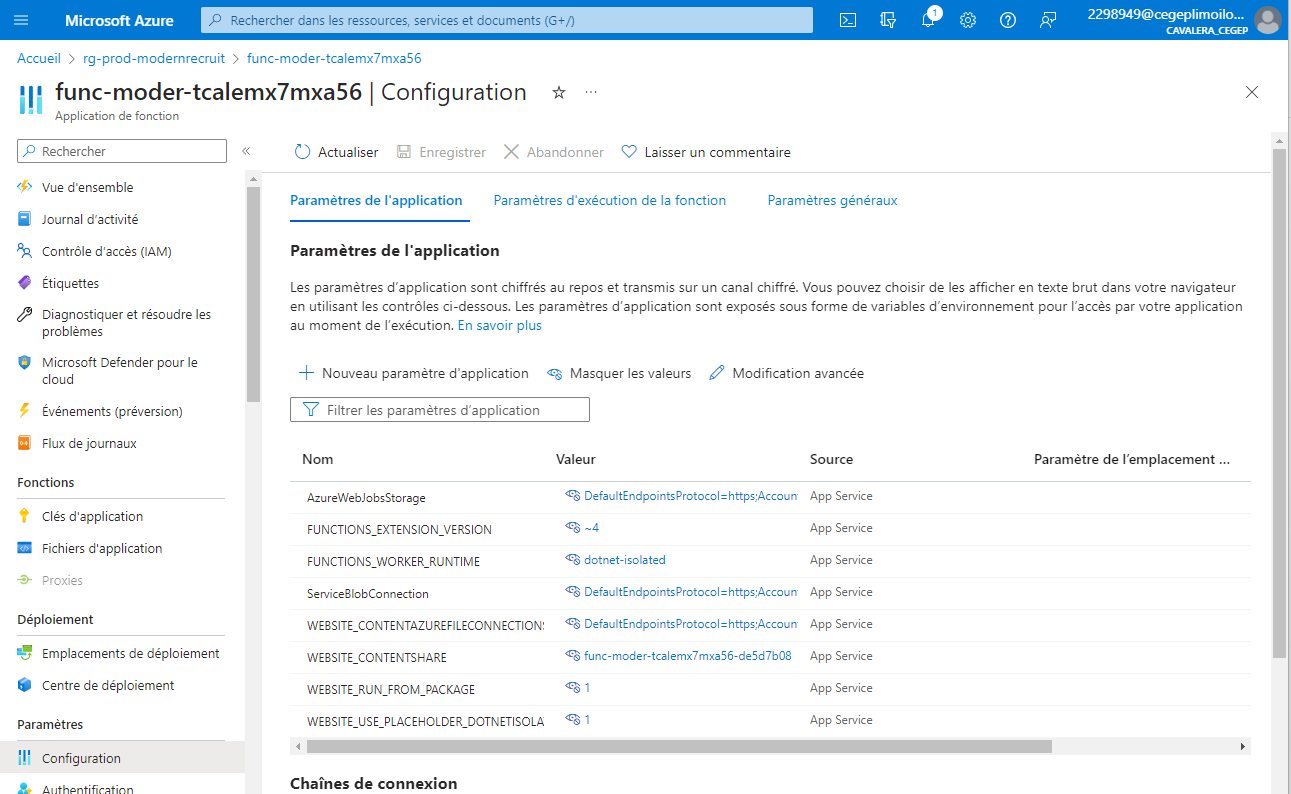




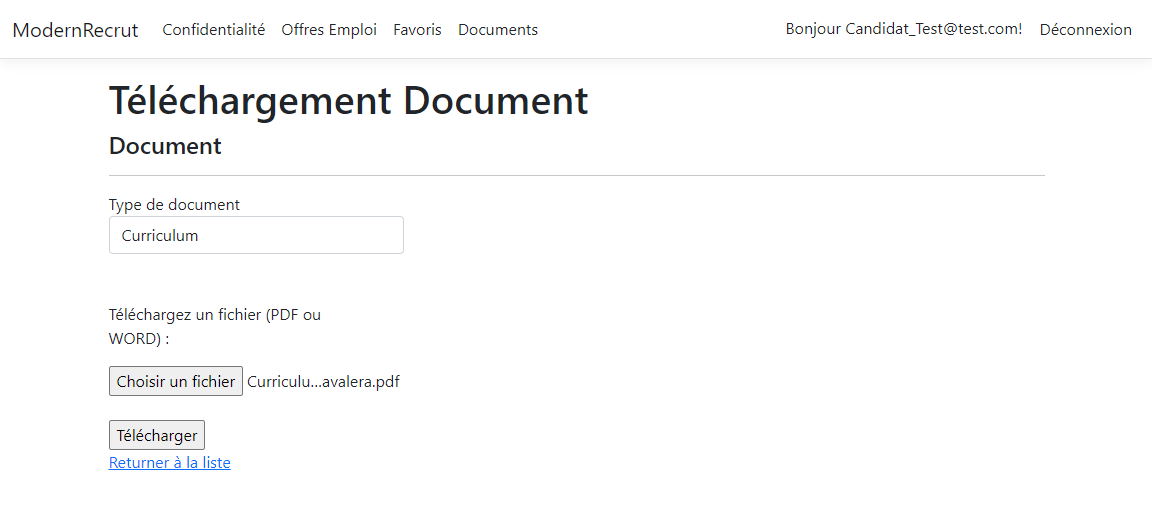
Copier la clé d’accès du compte de stockage pour la mettre dans la app function de Visual Studio.



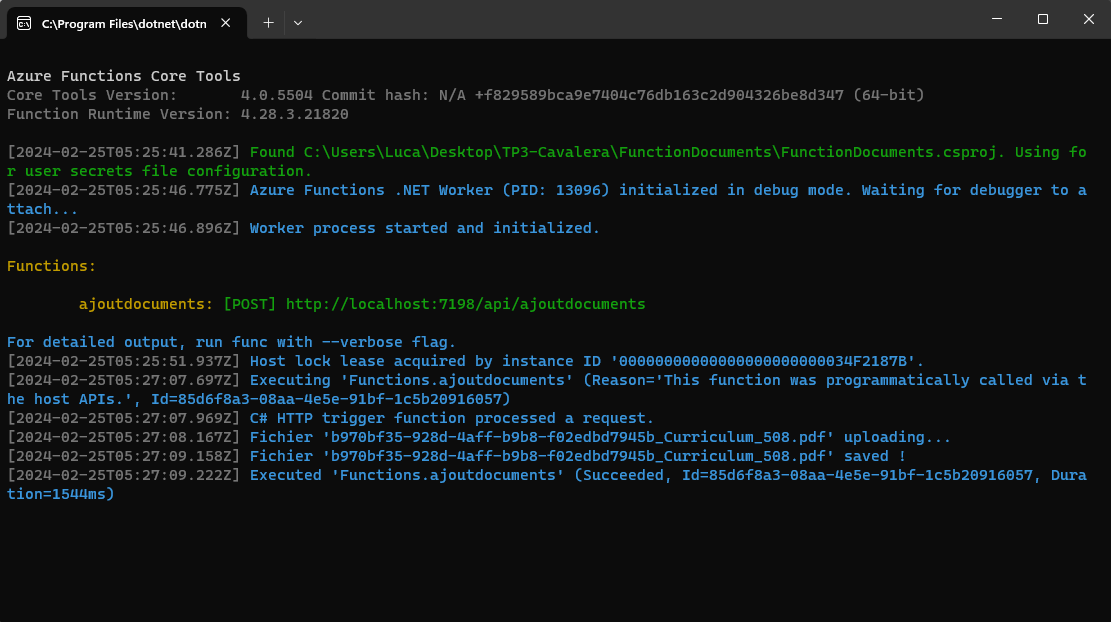
Ajout de la chaine de connexion du compte de stockage du fichier « local.settings.json » de VS dans l’app function de Azure.



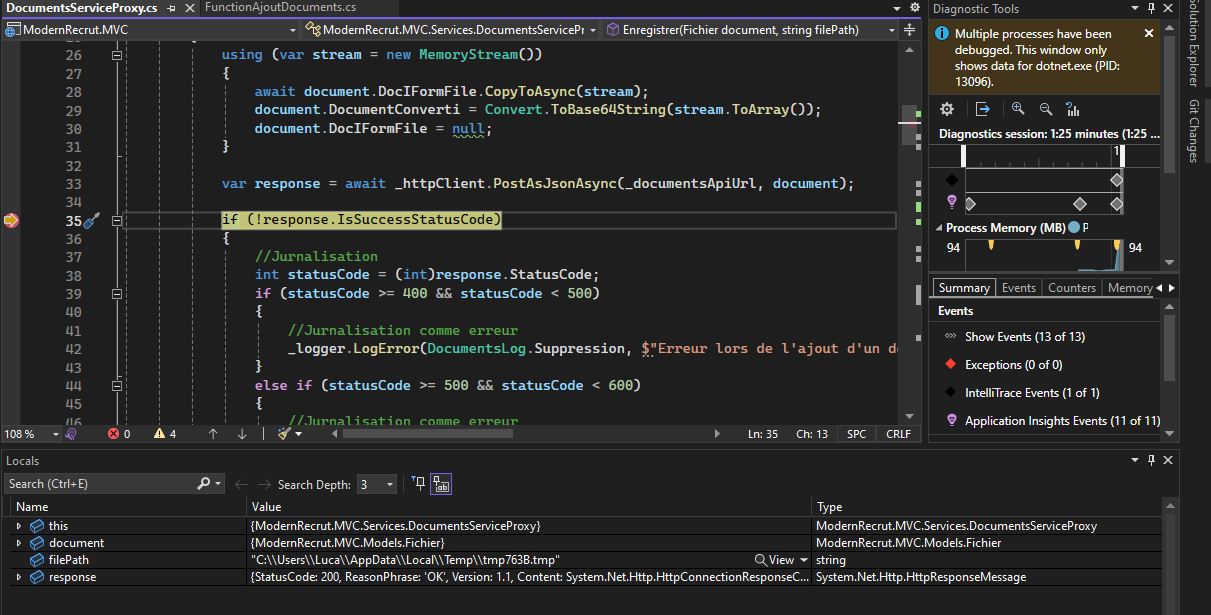
Téléchargement du document dans l’application MVC.



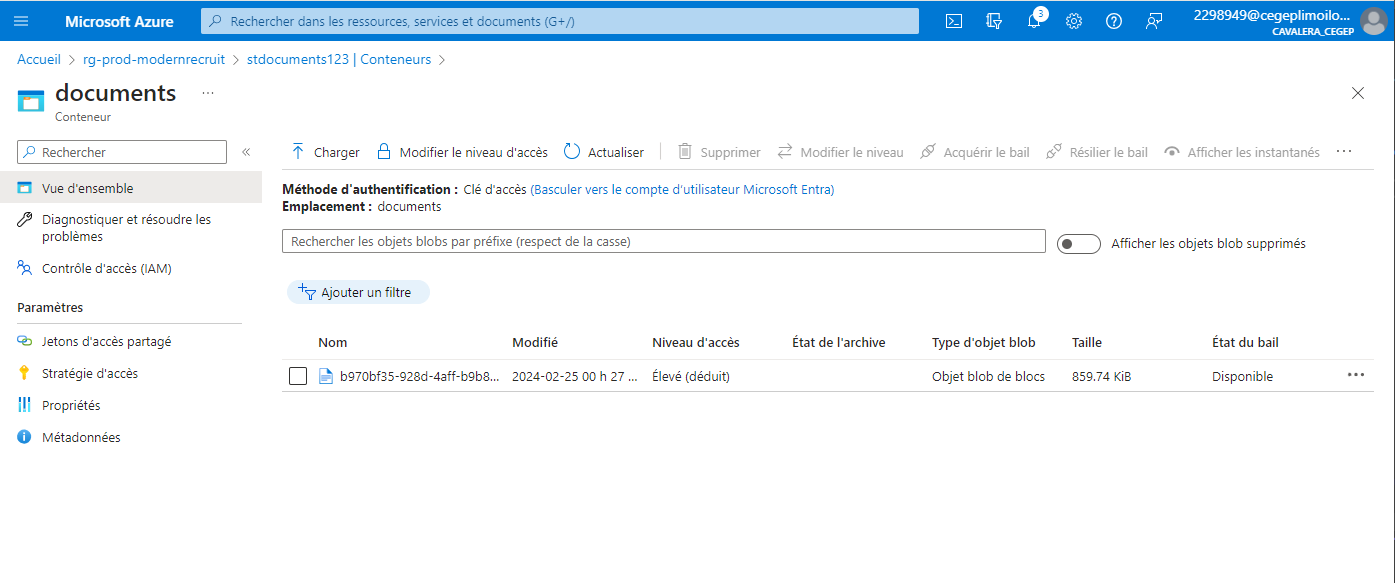
Exécution de la fonction.



Réponse Http – code 200.

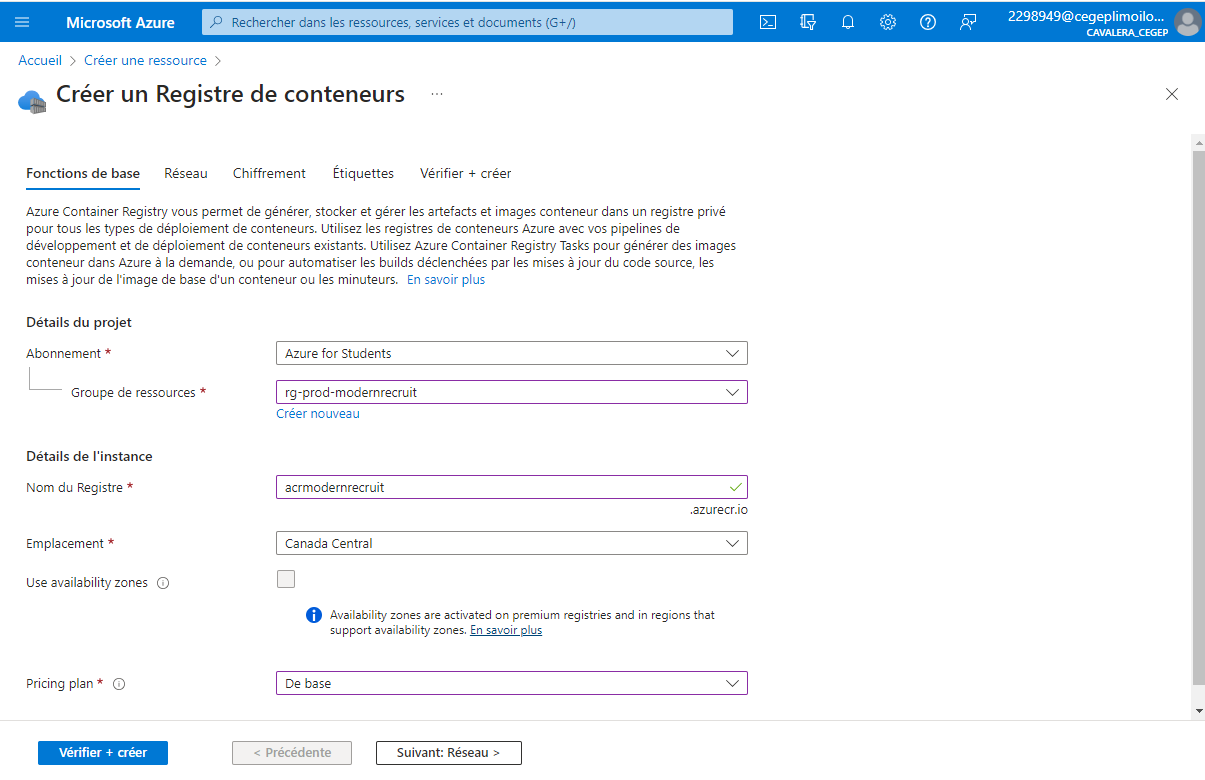


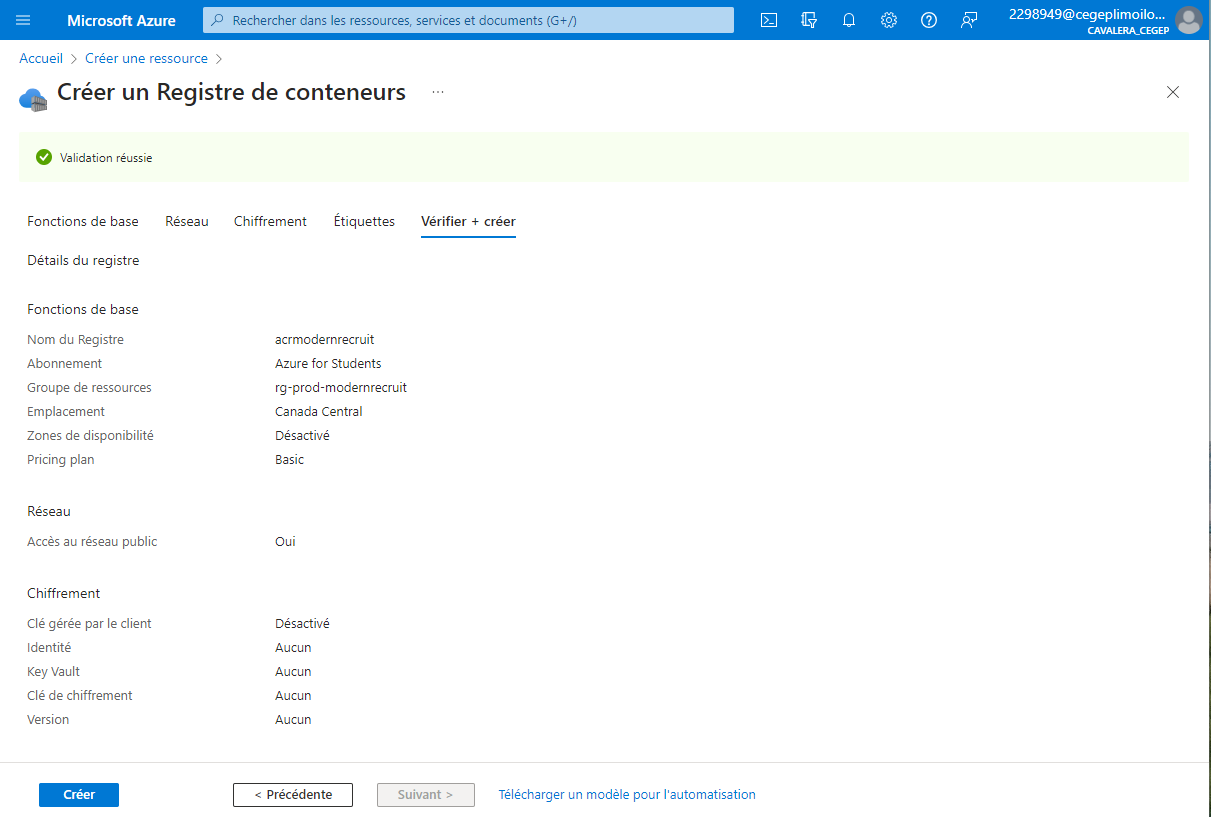
Chargement du fichier dans le compte de stockage suite à l’exécution de l’App Function.

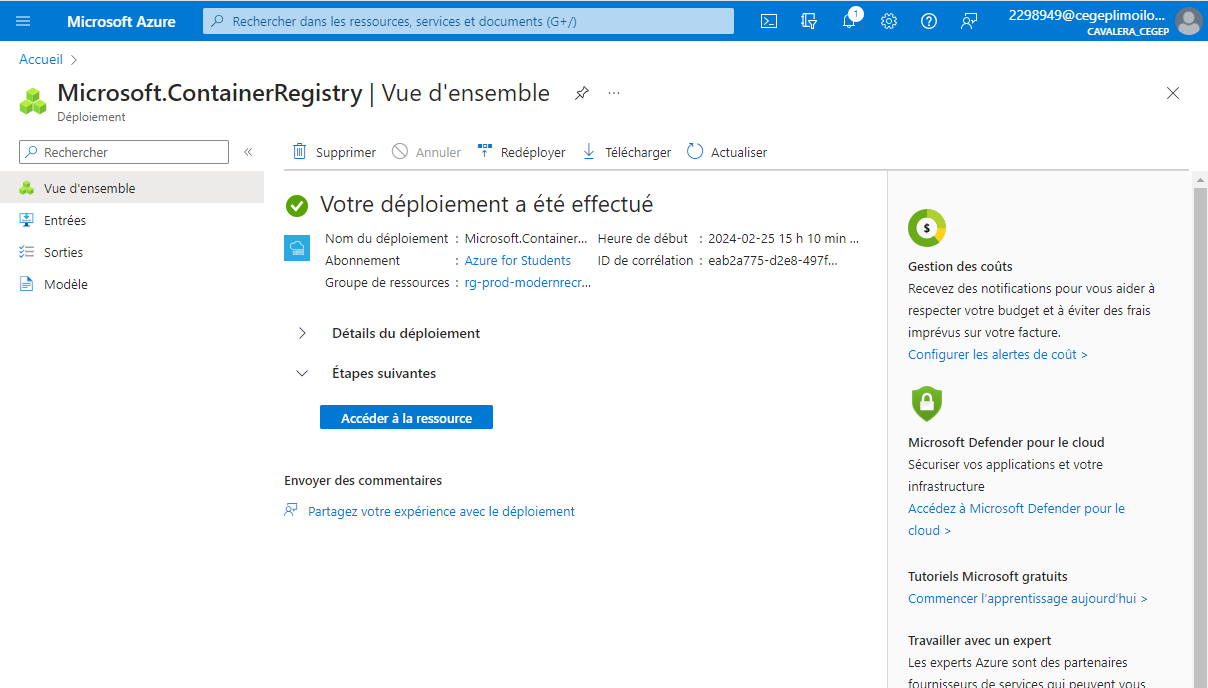


**Exercice4 :**

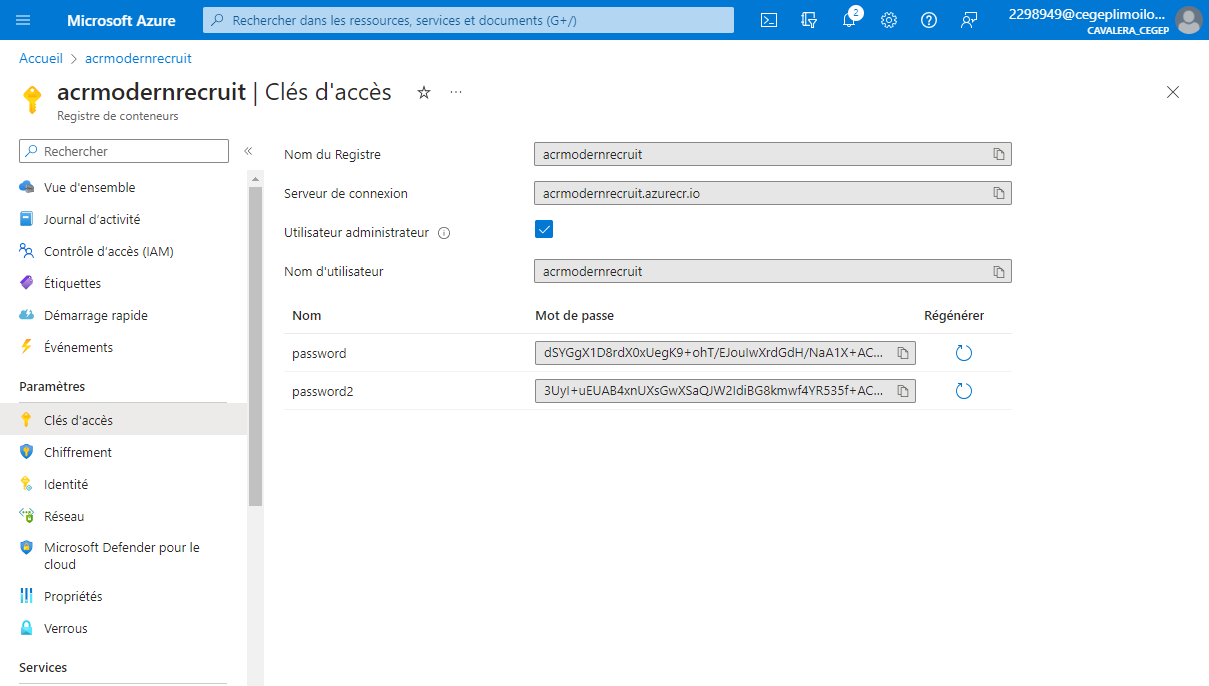
Création du registre de conteneurs.



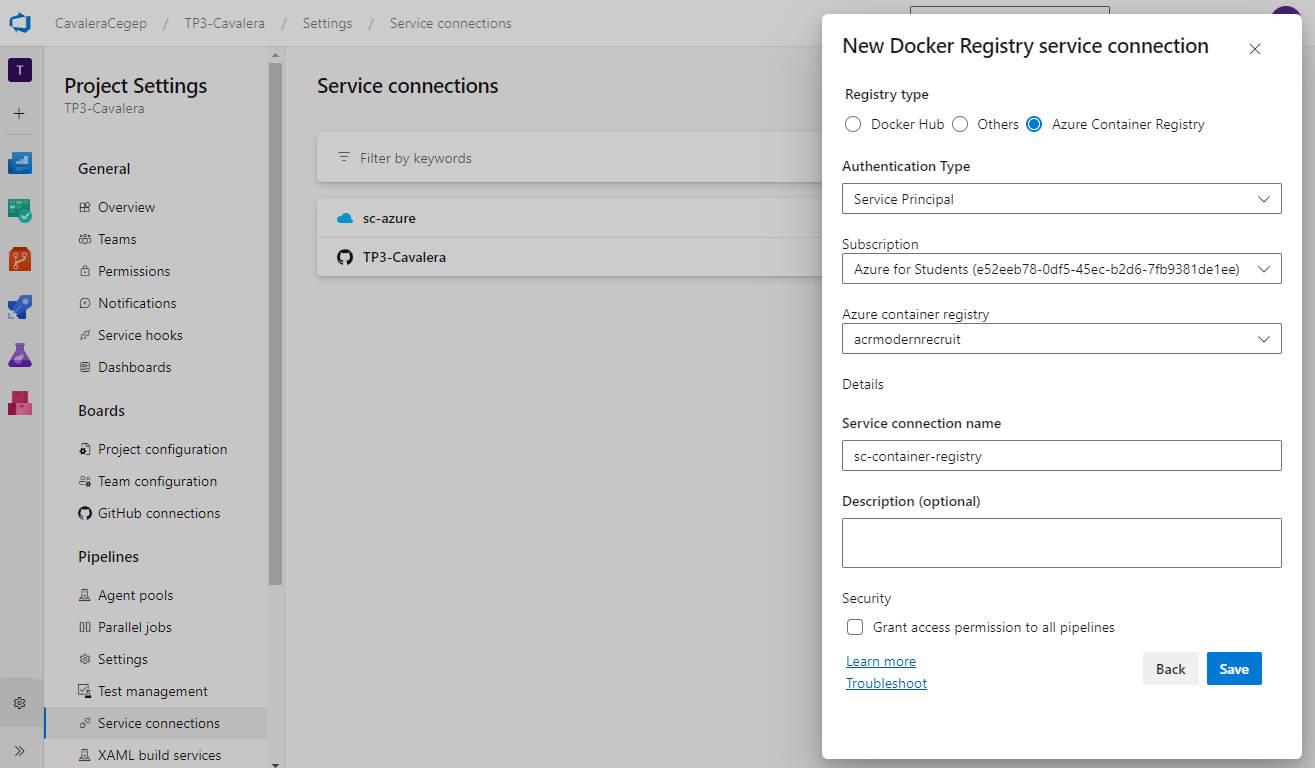


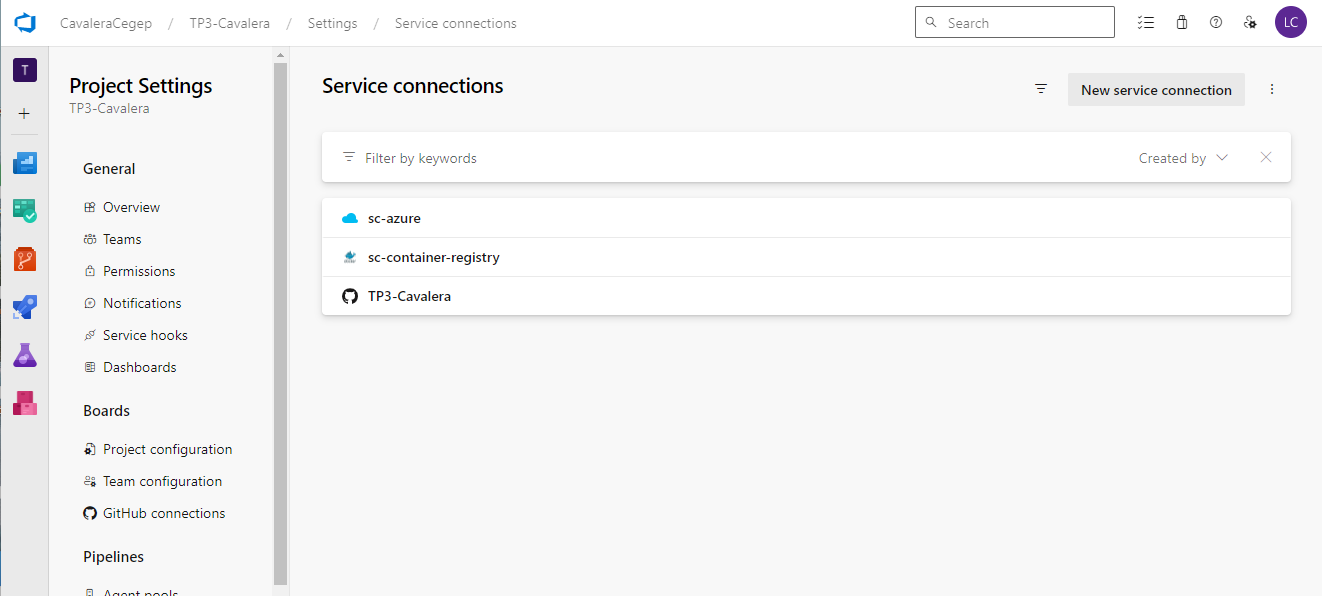


Activation de l’utilisateur administrateur.

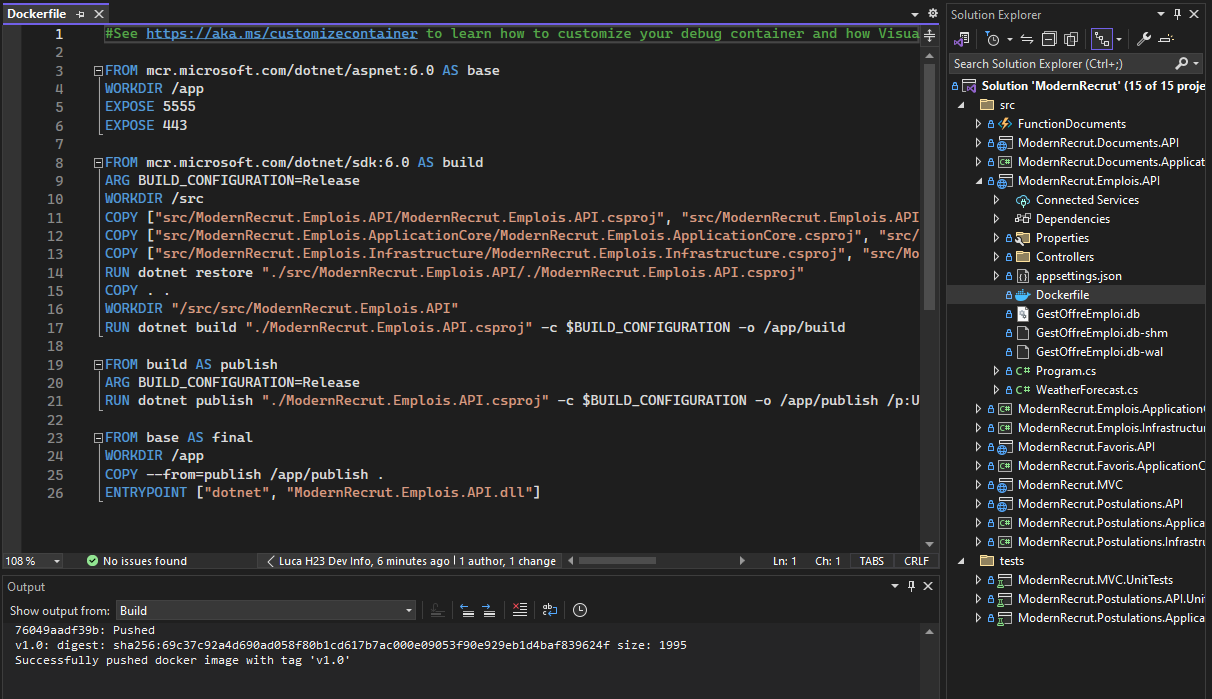


Création du service connexion dans Azure DevOps pour se connecter au registre de conteneurs.





Configuration de l’api emplois avec le service de conteneurs Docker.



Exécution du pipeline permettant de générer et déployer l’api Emplois dans Azure Container Instance.

