

Trabajo Práctico 9

Implementación de Contenedores en Azure Parte 2

4.1 Modificar nuestro pipeline para incluir el deploy en QA y PROD de Imágenes Docker en Servicio Azure App Services con Soporte para Contenedores

4.1.1 - Agregar a nuestro pipeline una nueva etapa que dependa de nuestra etapa de Construcción y Pruebas y de la etapa de Construcción de Imágenes Docker y subida a ACR realizada en el TP08

Agregar tareas para crear un recurso Azure Container Instances que levante un contenedor con nuestra imagen de back utilizando un AppServicePlan en Linux

Microsoft Azure

Inicio >

Crear plan de App Service

Datos básicos Etiquetas Revisar y crear

Resumen

Plan de App Service
de Microsoft

Detalles

Suscripción	Azure subscription 1
Grupo de recursos	TPSIngSoft3UCC2024
Nombre	MiAppPlan02
Sistema operativo	Linux
Región	Brazil South
SKU	Gratis
ACU	Infraestructura compartida
Memoria	1 GB de memoria

[Crear](#) [< Anterior](#) [Siguiente >](#) [Descargar una](#)

Microsoft Azure

Inicio >

Microsoft.Web-ASP-Portal-036b7844-af87 | Información general

Implementación

Buscar

[Eliminar](#) [Cancelar](#) [Volver a implementar](#) [Descargar](#) [Actualizar](#)

Se completó la implementación

Nombre de implementación : Microsoft.Web-ASP-Portal-036b7... Hora de inicio : 19/10/2024, 11:51:38
Suscripción : Azure subscription 1 Id. de correlación : 16fc7ff2-314c-4a61-8fe2-e666abdd633d
Grupo de recursos : TPSIngSoft3UCC2024

[Detalles de implementación](#)

[Pasos siguientes](#)

[Ir al recurso](#)

Impl
La imple
036b784-
recursos
[Ir al r](#)

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

Copilot

Inicio > Microsoft.Web-ASP-Portal-036b7844-af87 | Información general >

MiAppPlan02

Plan de Linux

Buscar

Eliminar

Envíenos sus comentarios

Introducción

Registro de actividad

Control de acceso (IAM)

Etiquetas

Diagnosticar y solucionar problemas

Eventos (versión preliminar)

Configuración

Supervisión

Automation

Ayuda

Las métricas del plan App Service no se aplican a las SKU gratuitas y compartidas

Información esencial

Grupo de recursos (mover) : TPSIngSoft3UCC2024

Estado : Listo

Ubicación : Brazil South

Suscripción (mover) : Azure subscription 1

Id. de suscripción : 128aebbd-e86d-40f6-9380-425f4e5c024e

Etiquetas (editar) : Agregar etiquetas

Plan de precios : F1

Recuento de instancias : 1

Aplicaciones y espacios : 0 / 0

Sistema operativo : Linux

Con redundancia de zona : Deshabilitado

Porcentaje de CPU

Porcentaje de memoria

Datos de entrada

46

47

WebAppApiNameContainersQA: 've-crud-api-qa'

AppServicePlanLinux: MiAppPlan02

#20241020.2 • Update azure-pipelines.yml for Azure Pipelines

Run new

This run is being retained as one of 3 recent runs by main (Branch).

View retention leases

Summary

Environments

Associated pipelines

Code Coverage

Manually run by Victoria Elliott

View change

Repository and version

Time started and elapsed

Related

Tests and coverage

tp7-angular

Today at 0:16

0 work items

Get started

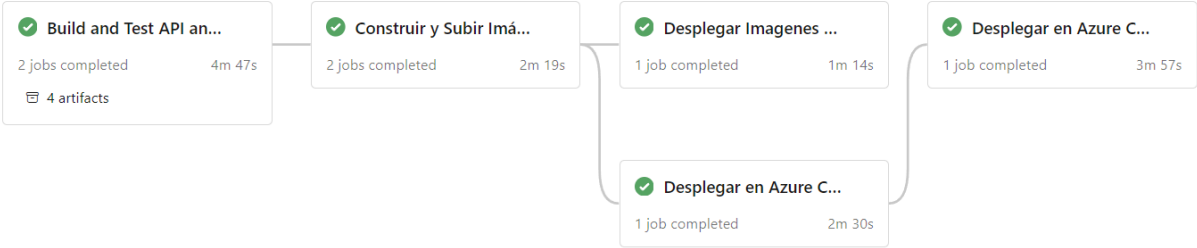
main 3c52faca

28m 32s

4 published

Stages

Jobs



Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Copilot

vikielliott30@gmail.com
DIRECTORIO PREDETERMINADO

Inicio > MiAppPlan02

MiAppPlan02 | Aplicaciones

Plan de Linux

Buscar

Actualizar

Filtrar por nombre

Estado: Todo

Introducción

Registro de actividad

Control de acceso (IAM)

Etiquetas

Diagnosticar y solucionar problemas

Fuente: función malintencionada

Nombre

Tipo

Variante

Grupo de recursos

Estado

ve-crud-api-qa

Aplicación

app,linux,container

TPSingSoft3UCC2024

En ejecución

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Copilot

vikielliott30@gmail.com
DIRECTORIO PREDETERMINADO

Inicio >

ve-crud-api-qa

Aplicación web

Buscar

Examinar

Detener

Intercambiar

Reiniciar

Eliminar

Actualizar

Descargar perfil de publicación

Restablecer perfil de publicación

Introducción

Registro de actividad

Control de acceso (IAM)

Etiquetas

Diagnosticar y solucionar problemas

Microsoft Defender for Cloud

Eventos (versión preliminar)

Mejor juntos (versión preliminar)

Implementación

Configuración

Rendimiento

Plan de App Service

Essentials

Grupo de recursos (...): TPSingSoft3UCC2024

Dominio predeterminado: ve-crud-api-qa.azurewebsites.net

Estado: En ejecución

Plan del servicio de a...: MiAppPlan02 (F1: 1)

Ubicación (mover): Brazil South

Sistema operativo: Linux

Suscripción (mover): Azure subscription 1

Comprobación de est...: Sin configurar

Id. de suscripción: 128aebbd-e86d-40f6-9380-425f4e5c024e

Etiquetas (editar): Agregar etiquetas

Propiedades

Supervisión

Registros

Funcionalidades

Notificaciones

Recomendaciones

Aplicación web

Nombre: ve-crud-api-qa

Modelo de publicación: Contenedor

Imagen de contenedor: veingsoft3.azurecr.io/employee-crud-apilatest

No es seguro

ve-crud-api-qa.eastus.azurecontainer.io/api/employee/getall

Dar formato al texto

[{"id":1,"name":"Juan Perez","createdAt":"2024-09-05T00:00:00"}, {"id":2,"name":"Carla Ruiz","createdAt":"2024-09-05T22:22:47.440572"}, {"id":3,"name":"Carlos Gomez","createdAt":"2024-09-06T09:16:50.870943"}, {"id":4,"name":"Joaquin Zarate","createdAt":"2024-09-06T09:19:37.898716"}, {"id":5,"name":"Luis Rodriguez","createdAt":"2024-09-06T09:23:31.324434"}, {"id":6,"name":"VictoriaE","createdAt":"2024-10-13T18:11:51.4051098"}, {"id":7,"name":"Victoria ","createdAt":"2024-10-13T18:12:16.8090433"}, {"id":8,"name":"VictoriaE ","createdAt":"2024-10-13T18:16:28.5401491"}]

4.2 Desafíos:

- 4.2.1 Agregar tareas para generar Front en Azure App Service con Soporte para Contenedores

```
48 | WebAppFrontNameContainersQA: 've-crud-front-qa'
```

main tp7-angular / azure-pipelines.yml *

```
435 |
436 |         Settings
437 |         -- task: AzureCLI@2
438 |         displayName: 'Verificar y crear el recurso Azure App Service para FRONT si no existe'
439 |         inputs:
440 |           azureSubscription: '$(ConnectedServiceName)'
441 |           scriptType: 'bash'
442 |           scriptLocation: 'inlineScript'
443 |           inlineScript: |
444 |             # Verificar si el App Service para la FRONT ya existe
445 |             if ! az webapp list --query "[?name=='$(WebAppFrontNameContainersQA)' && resourceGroup=='$(ResourceGroupName)']" --length 0; then
446 |               echo "El App Service para FRONT no existe. Creando..."
447 |               # Crear el App Service sin especificar la imagen del contenedor
448 |               az webapp create --resource-group $(ResourceGroupName) --plan $(AppServicePlanLinux) --name $(WebAppFrontNameContainersQA)
449 |             else
450 |               echo "El App Service para FRONT QA ya existe. Actualizando la imagen..."
451 |             fi
452 |             # Configurar el App Service para usar Azure Container Registry (ACR)
453 |             az webapp config container set --name $(WebAppFrontNameContainersQA) --resource-group $(ResourceGroupName) \
454 |               --container-image-name $(acrLoginServer)/$(frontImageName):$(frontImageTag) \
455 |               --container-registry-url https://$(acrLoginServer) \
456 |               --container-registry-user $(acrName) \
457 |               --container-registry-password $(az acr credential show --name $(acrName) --query "passwords[0].value" --o tsv)
458 |             # Establecer variables de entorno
459 |             az webapp config appsettings set --name $(WebAppFrontNameContainersQA) --resource-group $(ResourceGroupName) \
460 |               --settings ConnectionStrings_DefaultConnection="$(cnn-string-qa)"
461 |
462 |
```

- 4.2.2 Agregar variables necesarias para el funcionamiento de la nueva etapa considerando que debe haber 2 entornos QA y PROD para Back y Front.

Variables



api_url_prod

= http://ve-crud-api-prod.eastus.azurecontainer.io/api/em...



api_url_qa

= http://ve-crud-api-qa.eastus.azurecontainer.io/api/emplo...



cnn-string-prod



cnn-string-qa

```

--memory=$(container-memory-front-prod)
#-----
### STAGE DEPLOY TO AZURE APP SERVICE QA
#-----
- stage: DeployImagesToAppServiceQA
  displayName: 'Desplegar Imagenes en Azure App Service (QA)'
  dependsOn:
    - BuildAndTestBackAndFront
    - DockerBuildAndPush
  condition: succeeded()
  jobs:
    - job: DeployImagesToAppServiceQA
      displayName: 'Desplegar Imagenes de API y Front en Azure App Service (QA)'
      pool:
        vmImage: 'ubuntu-latest'
      steps:
        #-----
        # DEPLOY DOCKER API IMAGE TO AZURE APP SERVICE (QA)
        #-----
        Settings
        -- task: AzureCLI@2
          displayName: 'Verificar y crear el recurso Azure App Service para API (QA) si no existe'
          inputs:
            azureSubscription: '$(ConnectedServiceName)'
            scriptType: 'bash'
            scriptLocation: 'inlineScript'
            inlineScript: |
              # Verificar si el App Service para la API ya existe
              if ! az webapp list --query "[?name=='$(WebAppApiNameContainersQA)'] && resourceGroup==" \> /dev/null; then
                echo "El App Service para API QA no existe. Creando..."
                # Crear el App Service sin especificar la imagen del contenedor
                az webapp create --resource-group $(ResourceGroupName) --plan $(AppServicePlanLinux)
              else
                echo "El App Service para API QA ya existe. Actualizando la imagen..."
              fi
```

- **4.2.3 Agregar tareas para correr pruebas de integración en el entorno de QA de Back y Front creado en Azure App Services con Soporte para Contenedores.**
- **4.2.4 Agregar etapa que dependa de la etapa de Deploy en QA que genere un entorno de PROD.**

```
50 ··WebAppApiNameContainersProd: 've-crud-api-prod'
51 ··WebAppFrontNameContainersProd: 've-crud-front-prod'
```

← tp7-angular

```
main ▾ tp7-angular / azure-pipelines.yml *

465
466 --stage: DeployImagesToAppServiceProd
467 --displayName: 'Desplegar Imágenes en Azure App Service (Prod)'
468 --dependsOn:
469 --  - DeployImagesToAppServiceQA
470 --condition: succeeded()
471 --jobs:
472 --  - job: DeployImagesToAppServiceProd
473 --  --displayName: 'Desplegar Imágenes de API y Front en Azure App Service (Prod)'
474 --  --pool:
475 --  --vmImage: 'ubuntu-latest'
476 --  --steps:
477 --  --  Settings
478 --  --  --task: AzureCLI@2
479 --  --  --displayName: 'Verificar y crear el recurso Azure App Service para API (Prod) si no existe'
480 --  --  --inputs:
481 --  --  --  azureSubscription: '$(ConnectedServiceName)'
482 --  --  --  scriptType: 'bash'
483 --  --  --  scriptLocation: 'inlineScript'
484 --  --  --  inlineScript: |
485 --  --  --  # Verificar si el App Service para la API ya existe
486 --  --  --  if ! az webapp list --query "[?name=='$(WebAppApiNameContainersProd)']" --resource-group=$(ResourceGroup)
487 --  --  --  echo "El App Service para API Prod no existe. Creando..."
488 --  --  --  # Crear el App Service sin especificar la imagen del contenedor
489 --  --  --  az webapp create --resource-group $(ResourceGroupName) --plan $(AppServicePlanLinux) --name $(WebAppApi
490 --  --  --  else
491 --  --  --  echo "El App Service para API Prod ya existe. Actualizando la imagen..."
492 --  --  --  fi
```

← tp7-angular

```
main ▾ tp7-angular / azure-pipelines.yml *

502 --settings: ConnectionStrings__DefaultConnection="$(cnn-string-prod)"
503
504 --  Settings
505 --  --task: AzureCLI@2
506 --  --displayName: 'Verificar y crear el recurso Azure App Service para FRONT si no existe'
507 --  --inputs:
508 --  --  azureSubscription: '$(ConnectedServiceName)'
509 --  --  scriptType: 'bash'
510 --  --  scriptLocation: 'inlineScript'
511 --  --  inlineScript: |
512 --  --  # Verificar si el App Service para la FRONT ya existe
513 --  --  if ! az webapp list --query "[?name=='$(WebAppFrontNameContainersProd)']" --resource-group=$(ResourceGroupName) | .
514 --  --  echo "El App Service para FRONT no existe. Creando..."
515 --  --  # Crear el App Service sin especificar la imagen del contenedor
516 --  --  az webapp create --resource-group $(ResourceGroupName) --plan $(AppServicePlanLinux) --name $(WebAppFrontNameContai
517 --  --  else
518 --  --  echo "El App Service para FRONT Prod ya existe. Actualizando la imagen..."
519 --  --  fi
520 --  --  # Configurar el App Service para usar Azure Container Registry (ACR)
521 --  --  az webapp config container set --name $(WebAppFrontNameContainersProd) --resource-group $(ResourceGroupName) \
522 --  --  --container-image-name $(acrLoginServer)/$(frontImageName):$(frontImageTag) \
523 --  --  --container-registry-url https://$(acrLoginServer) \
524 --  --  --container-registry-user $(acrName) \
525 --  --  --container-registry-password $(az acr credential show --name $(acrName) --query "passwords[0].value" -o tsv)
526 --  --  # Establecer variables de entorno
527 --  --  az webapp config appsettings set --name $(WebAppFrontNameContainersProd) --resource-group $(ResourceGroupName) \
528 --  --  --settings: ConnectionStrings__DefaultConnection="$(cnn-string-prod)"
529 --  --  --settings: ConnectionStrings__DefaultConnection="$(cnn-string-prod)"
530 --  --  --settings: ConnectionStrings__DefaultConnection="$(cnn-string-prod)"
```

✓ #20241020.5 • Update azure-pipelines.yml for Azure Pipelines

Run new

🏠 tp7-angular

This run is being retained as one of 3 recent runs by main (Branch).

View retention leases

Summary

Environments

Associated pipelines

Code Coverage

Manually run by Victoria Elliott

View change

Repository and version

Time started and elapsed

Related

Tests and coverage

🔗 tp7-angular

📅 Today at 1:55

🔗 0 work items

📄 Get started

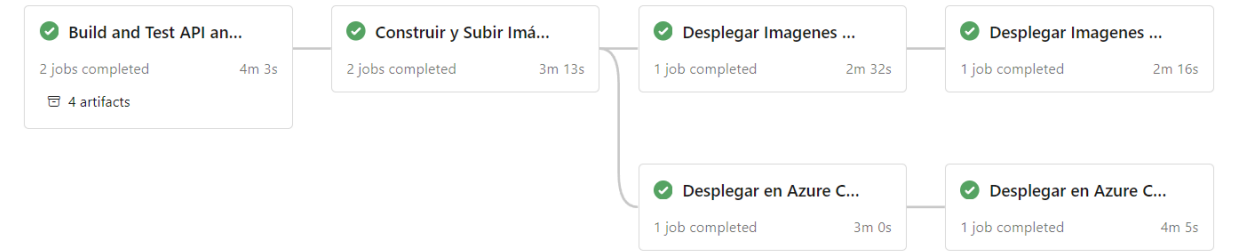
📁 main 🔗 311556b3

⌚ 40m 24s

📄 4 published

Stages

Jobs



Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

Copilot

🔔

⚙️

🔍

🗨️

vikielliott30@gmail.com

DIRECTORIO PREDETERMINADO

Inicio > MiAppPlan02

MiAppPlan02 | Aplicaciones

Plan de Linux

Buscar

Actualizar

Filtrar por nombre

Estado: Todo

Nombre	Tipo	Variante	Grupo de recursos	Estado
ve-crud-api-prod	Aplicación	app.linux.container	TPSIngSoft3UCC2024	En ejecución
ve-crud-api-qa	Aplicación	app.linux.container	TPSIngSoft3UCC2024	En ejecución
ve-crud-front-prod	Aplicación	app.linux.container	TPSIngSoft3UCC2024	En ejecución
ve-crud-front-qa	Aplicación	app.linux.container	TPSIngSoft3UCC2024	En ejecución

← → ↺

No es seguro

ve-crud-api-qa.eastus.azurecontainer.io/api/employee/getall

🔍

☆

👤


⋮

Dar formato al texto

[{"id":1,"name":"Juan Perez","createdAt":"2024-09-05T00:00:00"}, {"id":2,"name":"Carla Ruiz","createdAt":"2024-09-05T22:22:47.440572"}, {"id":3,"name":"Carlos Gomez","createdAt":"2024-09-06T09:10:50.070943"}, {"id":4,"name":"Joaquin Zarate","createdAt":"2024-09-06T09:19:37.898716"}, {"id":5,"name":"Luis Rodriguez","createdAt":"2024-09-06T09:23:31.324434"}, {"id":6,"name":"VictoriaE","createdAt":"2024-10-13T18:11:51.4051098"}, {"id":7,"name":"Victoria","createdAt":"2024-10-13T18:12:16.8090433"}, {"id":8,"name":"VictoriaE ","createdAt":"2024-10-13T18:16:28.5401491"}]

EmployeeCrudAngular

Employees:  [Add New Employee](#)

Id	Name	Created Date	
1	Juan Perez	05/09/2024 00:00:00	 
2	Carla Ruiz	05/09/2024 22:22:47	 
3	Carlos Gomez	06/09/2024 09:16:50	 
4	Joaquin Zarate	06/09/2024 09:19:37	 
5	Luis Rodriguez	06/09/2024 09:23:31	 
6	VictoriaE	13/10/2024 18:11:51	 
7	Victoria	13/10/2024 18:12:16	 
8	VictoriaE	13/10/2024 18:16:28	 

- **4.2.5 Entregar un pipeline que incluya:**
 - **A) Etapa Construcción y Pruebas Unitarias y Code Coverage Back y Front**
 - **B) Construcción de Imágenes Docker y subida a ACR**
 - **C) Deploy Back y Front en QA con pruebas de integración para Azure Web Apps**
 - **D) Deploy Back y Front en QA con pruebas de integración para ACI**
 - **E) Deploy Back y Front en QA con pruebas de integración para Azure Web Apps con Soporte para contenedores**
 - **F) Aprobación manual de QA para los puntos C,D,E**
 - **G) Deploy Back y Front en PROD para Azure Web Apps**
 - **H) Deploy Back y Front en PROD para ACI**
 - **I) Deploy Back y Front en PROD para Azure Web Apps con Soporte para contenedores**

Pipeline llamado “tp7-angular” en proyecto: “tp7-angular”