



Universidad de Guadalajara  
Centro Universitario  
de  
Ciencias Exactas  
e  
Ingenierías  
(CUCEI)

Materia: COMPUTACION TOLERANTE A FALLAS

Nombre:  
Jonathan Aceves López

Código:  
217481363  
NRC: 179961

1/10/2023

El siguiente informe analiza un fragmento de código Python que utiliza la biblioteca Prefect para crear y desplegar un flujo de trabajo. La biblioteca Prefect es una herramienta poderosa para la orquestación de flujos de trabajo en Python. En este caso, el código define un flujo de trabajo que ejecuta una función llamada `hello_world`, la cual imprime saludos y despedidas según los argumentos proporcionados. El propósito de este informe es proporcionar una comprensión más profunda del código y su funcionamiento.

Código:

```
from prefect import flow

@flow(log_prints=True)
def hello_world(name: str = "world", goodbye: bool = False):
    print(f"Hello {name} from Prefect!")

    if goodbye:
        print(f"Goodbye {name}!")

if __name__ == "__main__":
    hello_world.serve(name="my-first-deployment",
                      tags=["onboarding"],
                      parameters={"goodbye": True},
                      interval=60)
```

**Importación de Prefect:** El código comienza importando la función `flow` de la biblioteca Prefect, que se utiliza para definir flujos de trabajo.

**Definición de la Función `hello_world`:** Se define una función llamada `hello_world` que acepta dos argumentos: `name` (una cadena de texto) y `goodbye` (un valor booleano). La función está decorada con `@flow(log_prints=True)`, lo que indica que se trata de un flujo de trabajo Prefect y habilita el registro de impresiones.

**Saludos y Despedidas:** Dentro de la función `hello_world`, se imprimen saludos utilizando el valor de `name` y, si `goodbye` es `True`, también se imprime un mensaje de despedida.

**Llamada al Flujo de Trabajo:** El bloque `if __name__ == "__main__":` garantiza que el código debajo de él solo se ejecute cuando el archivo se ejecute directamente. Dentro de ese bloque, se llama a `hello_world.serve()` para desplegar el flujo de trabajo Prefect:

Se le asigna un nombre al flujo de trabajo: "my-first-deployment".

Se etiqueta el flujo de trabajo como "onboarding".

Se configura un valor de parámetro para goodbye, lo que significa que el flujo de trabajo se ejecutará con goodbye establecido en True.

Se especifica un intervalo de ejecución de 60 segundos, lo que significa que el flujo de trabajo se ejecutará cada minuto.

```
Your flow 'hello-world' is being served and polling for scheduled runs!

To trigger a run for this flow, use the following command:

$ prefect deployment run 'hello-world/my-first-deployment'

You can also run your flow via the Prefect UI:
https://app.prefect.cloud/account/3bae1c3f-87fb-4cdf-a8ba-920449a18f79/workspace/8e838844-f391-4c1a-b7c8-ff4e3da7c889/deployments/deployment/441cb78f-2703-4fea-8658-3c1a8d8f12b3

11:05:41.032 | INFO | prefect.flow_runs.runner - Runner 'my-first-deployment' submitting flow run '4aa02e78-23f9-4ce9-804c-d5b1889d5019'
11:05:41.171 | INFO | prefect.flow_runs.runner - Opening process...
11:05:41.186 | INFO | prefect.flow_runs.runner - Completed submission of flow run '4aa02e78-23f9-4ce9-804c-d5b1889d5019'
<frozen runpy>:128: RuntimeWarning: 'prefect.engine' found in sys.modules after import of package 'prefect', but prior to execution of 'prefect.engine'; this may result in unpredictable behaviour
11:05:45.211 | INFO | Flow run 'bipedal-bison' - Downloading flow code from storage at 'C:\Users\jonat'
11:05:45.964 | INFO | Flow run 'bipedal-bison' - Hello world from Prefect! \U0001f917
11:05:45.967 | INFO | Flow run 'bipedal-bison' - Goodbye world!
11:05:46.119 | INFO | Flow run 'bipedal-bison' - Finished in state Completed()
11:05:46.547 | INFO | prefect.flow_runs.runner - Process 13296 exited cleanly.
11:06:09.683 | INFO | prefect.flow_runs.runner - Runner 'my-first-deployment' submitting flow run 'c018e648-360b-4fa2-80da-73d75f038f85'
11:06:09.811 | INFO | prefect.flow_runs.runner - Opening process...
11:06:09.818 | INFO | prefect.flow_runs.runner - Completed submission of flow run 'c018e648-360b-4fa2-80da-73d75f038f85'
<frozen runpy>:128: RuntimeWarning: 'prefect.engine' found in sys.modules after import of package 'prefect', but prior to execution of 'prefect.engine'; this may result in unpredictable behaviour
11:06:13.266 | INFO | Flow run 'quantum-kingfisher' - Downloading flow code from storage at 'C:\Users\jonat'
11:06:18.161 | INFO | Flow run 'quantum-kingfisher' - Hello world from Prefect! \U0001f917
11:06:18.165 | INFO | Flow run 'quantum-kingfisher' - Goodbye world!
11:06:18.338 | INFO | Flow run 'quantum-kingfisher' - Finished in state Completed()
11:06:18.813 | INFO | prefect.flow_runs.runner - Process 13220 exited cleanly.
11:07:19.434 | INFO | prefect.flow_runs.runner - Runner 'my-first-deployment' submitting flow run '69a3735d-ae2b-4347-812f-5ba8d64c914c'
11:07:19.572 | INFO | prefect.flow_runs.runner - Opening process...
```



## Conclusión:

El código presentado demuestra cómo utilizar la biblioteca Prefect para crear y desplegar flujos de trabajo en Python. En este caso, el flujo de trabajo hello\_world imprime mensajes de saludo y despedida, lo que ilustra el potencial de Prefect para automatizar tareas y flujos de trabajo.

Prefect proporciona una solución versátil para la orquestación de flujos de trabajo, lo que facilita la automatización y programación de tareas en una variedad de aplicaciones. El código presentado aquí es un ejemplo simple de cómo comenzar a utilizar Prefect para gestionar flujos de trabajo.