cfa-cloudops

Python Package for Cloud Operations

Presenter: Ryan Raasch (xng3)

What is cfa-cloudops?

- open source python package
- provides easy interaction with the Cloud (Azure)
- single repository improving upon other existing Azure libraries
    - cfa_azure + cfa-azuretools

Motivation for the Refactor

- combine existing CFA Azure packages into a single standardized package
- easy authentication and development for data scientists
- updated python and dependent packages
- option for other cloud integrations
- workflow orchestration and model tracking
- part of greater `cfa.` namespace
    - `cfa.cloudops`
    - `cfa.dataops`

Why use cfa-cloudops

- easy authentication
- intuitive functions and workflows
- upgraded libraries
- more features beyond Azure integration coming soon

Azure API

```
from azure.batch import BatchServiceClient
from azure.batch.models import (
    PoolAddParameter, VirtualMachineConfiguration, ImageReference,
    DeploymentConfiguration, CloudServiceConfiguration, PoolLifetimeOption
)

# Replace with your Batch account details
BATCH_ACCOUNT_NAME = 'your_batch_account_name'
BATCH_ACCOUNT = 'your_batch_account_key'
BATCH_ACCOUNT_URL = 'your_batch_account_url'

# Create a Batch service client
```

```
batch_client = BatchServiceClient(
    batch_url=BATCH_ACCOUNT_URL,
    credentials=BatchSharedKeyCredentials(
        BATCH_ACCOUNT_NAME, BATCH_ACCOUNT_KEY
    )
)

pool_id = "my-python-pool"
vm_size = "Standard_A1_v2"  # Choose an appropriate VM size
target_dedicated_nodes = 1

# Define the virtual machine configuration (e.g., Ubuntu Server)
vm_configuration = VirtualMachineConfiguration(
    image_reference=ImageReference(
        publisher="Canonical",
        offer="UbuntuServer",
        sku="18.04-LTS",
        version="latest"
    ),
    node_agent_sku_id="batch.node.ubuntu 18.04"
)

# Create the pool
new_pool = PoolAddParameter(
    id=pool_id,
    vm_size=vm_size,
    target_dedicated_nodes=target_dedicated_nodes,
    virtual_machine_configuration=vm_configuration
)

batch_client.pool.add(new_pool)
print(f"Pool '{pool_id}' created successfully.")
```

cfa-cloudops

```
from cfa.cloudops import CloudClient

client = CloudClient()
client.create_pool(
    pool_name = "my-python-pool",
    container_name = "ubuntu:22.04"
)
```

Example creating a pool

Available Cloud Integrations

- Azure Batch
- Azure Container Registry

- Azure Blob Storage
- Azure Container App Jobs

Authentication Methods

- uses .env files or environment variables for more secure storing
- Managed Identity
  - less tokens/secrets to maintain locally
  - easier/convenient
- Service Principals
- Federated Token Credential (for GH Actions)

`cfa-cloudops` modules

- low level functions (similar to cfa-azuretools)
- CloudClient (similar to cfa_azure AzureClient)
- ContainerAppClient
- automation (run tasks from toml)
- local (debugging or initial development locally emulating Cloud environment)

Live Demo

Roadmap

- CLI commands: easy cloud interactions for any programming language
- Metaflow
- DAGster: workflow orchestrator

Documentation

https://cdcgov.github.io/cfa-cloudops/

Questions?

For more information or help getting started: Contact: Ryan Raasch (xng3@cdc.gov)
