ABSTRACT

As organizations increasingly embrace cloud computing, platforms like IBM Cloud Foundry provide a robust environment for deploying and managing applications. This abstract outlines the integration of Python scripts with IBM Cloud Foundry, emphasizing the use of serverless functions to enhance scalability and efficiency.

The proposed solution involves developing Python scripts tailored for execution within a Cloud Foundry environment. Leveraging the serverless paradigm, functions are designed to handle specific tasks, promoting modularity and ease of maintenance. IBM Cloud Foundry's versatility in supporting various programming languages, including Python, makes it an ideal choice for hosting and executing these script

PROGRAM

```
app = Flask(__name__)

def hello():

    return 'Hello, welcome to the IBM Cloud Foundry App!'



# Serverless function

@app.route('/invoke-function', methods=['POST'])

def invoke_function():

    data = request.json

    if 'action' not in data:

        return jsonify({'error': 'Missing "action" parameter'}), 400



    action = data['action']

    result = execute_serverless_function(action)
    return jsonify({'result': result})


def execute_serverless_function(action):
    # Call the IBM Cloud Function (OpenWhisk)
    openwhisk_url = 'https://openwhisk.ng.bluemix.net/api/v1/web/{namespace}/default/{action}.json'
    namespace = 'your-namespace'

    response = requests.post(
        openwhisk_url.format(namespace=namespace, action=action),
        json={'payload': 'your_payload'}
    )

    if response.status_code == 200:
        return response.json()
    else:
        return {'error': f'Function execution failed with status code {response.status_code}'}


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=int(port), debug=True)
```