GoDeploy: Portable Deployment of Serverless Functions in Federated FaaS

Sashko Ristov, Simon Brandacher, Michael Felderer, Ruth Breu

This paper is published in the IEEE Cloud Summit 2022 and achieved the best paper award.

Abstract: Federated Function-as-a-Service (FaaS) offers higher scalability, better resilience and cost-performance trade-off than running serverless applications in a single cloud region. However, existing Infrastructure-as-Code (IaC) tools are mainly focused on the FaaS provider, rather than on applications, which increases developer effort to code multiple times the same data in order to deploy a serverless function on various cloud regions in federated FaaS. To bridge this gap, this paper introduces *GoDeploy*, a framework that simplifies coding the deployment of serverless functions in Federated FaaS. Using the design principle "*code once, deploy everywhere*", *GoDeploy* offers developers a domain-specific language, which introduces a three-levels hierarchy serverless function → FaaS providers → cloud regions of FaaS provider, rather than existing either the two-levels hierarchy FaaS provider → serverless functions or flat horizontal structure. Moreover, *GoDeploy* hides the complexity and requirements of each FaaS provider to store deployment packages (zip) of serverless functions on their storages. With this approach, *GoDeploy* reduces deployment script length measured in lines of code (LoC) compared to the recent FaaSifier M2FaaS by up to 33.33% for deployment on three cloud regions of AWS. When deploying a single function on three cloud regions of each of three FaaS providers AWS, IBM, Google, LoC are reduced by up to 72.34% compared to the state-of-the-art IaC tool Terraform. The improvement is higher when a serverless function needs to be deployed on multiple cloud regions because *GoDeploy*'s three-level hierarchy requires a single LoC per cloud region, compared to multiple LoC in Terraform's and M2FaaS DSLs.

Publication: S. Ristov, S. Brandacher, M. Felderer and R. Breu, "GoDeploy: Portable Deployment of Serverless Functions in Federated FaaS,"2022 *IEEE Cloud Summit*, Fairfax, VA, USA, 2022, pp. 38-43, doi: 10.1109/CloudSummit54781.2022.00012.

1 *GoDeploy* system architecture

GoDeploy comprises four modules DSL, URL parser, federated storage interface, and federated deployer (Fig. 1). *GoDeploy* parses the input file that is written using the *GoDeploy* Domain Specific Language (DSL) to determine which functions should be deployed. *GoDeploy* requires a single copy of each function, e.g., *f1.zip* in AWS Virginia or locally. The *URL parser* determines the target providers and uses their SDKs to copy *f1.zip*. Finally, *federated deployer* deploys *f1.zip*.

While Terraform and Serverless framework put the provider at the top of the hierarchy, *GoDeploy* uses the function (deployment package), then FaaS providers, and their regions at the bottom (Fig. 2).

doi:10.18420/sw2024_09

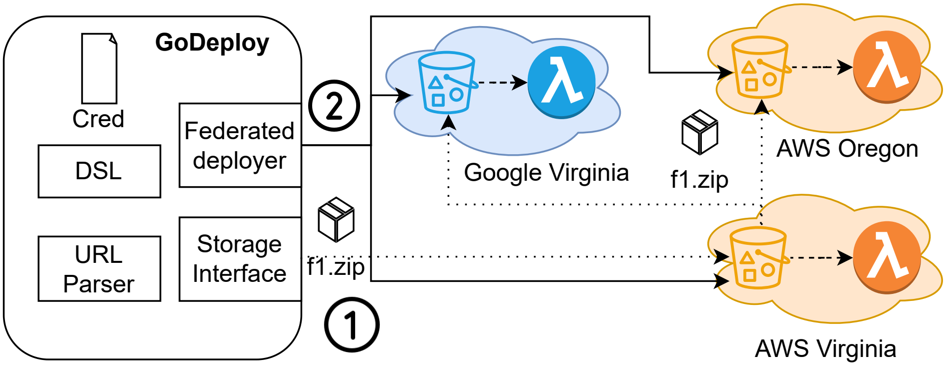

Abb. 1: *GoDeploy* architecture.

```
functions:
  - archive: "https://bucket.s3..."
    name: "f1"
    memory: 128
    timeout: 60
    providers:
      - name: "AWS"
        handler: "example.AWSHandler"
        regions:
          - "us-east-1"
          - "us-west-2"
        runtime: "go1.x"
      - name: "Google"
        handler: "example.GoogleHandler"
        regions:
          - "us-central1"
        runtime: "go116"
  - archive: "https://storage.cloud.google..."
    name: "f2"
    ...
```

Abb. 2: Example of deployment script in *GoDeploy* DSL.

2 Contribution

With the three-layered DSL, *GoDeploy* reduces lines of code by 33.33% compared to M2FaaS and 64.71% compared to Terraform. Even higher effect is achieved when a function needs to be deployed on multiple cloud regions of different providers. *GoDeploy* reduces the lines of code by 51.85% and 72.34%, compared to M2FaaS and Terraform, respectively.