

Terraform Actions

in Action

Mattias Fjellström



New in

Terraform 1.14

```
$ terraform version
```

```
Terraform v1.14.0
```

Terraform Core Workflow

Before Terraform 1.14

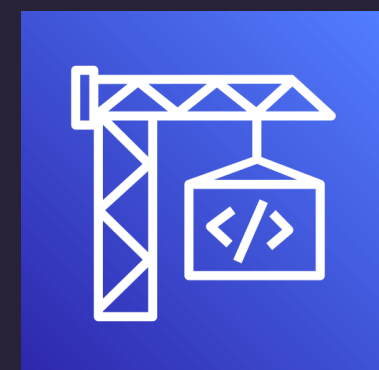
DELETE

CREATE

READ

UPDATE

```
+ resource "aws_s3_bucket" "demo" {  
+   bucket_prefix = "demo"  
+  
+   resource "aws_s3_bucket" "demo" {  
+     tags = {  
+       prefix = "demo"  
+       ManagedBy = "terraform"  
+       Owner      = "HUG"  
+     }  
+   }  
+ }
```



Terraform
workflow



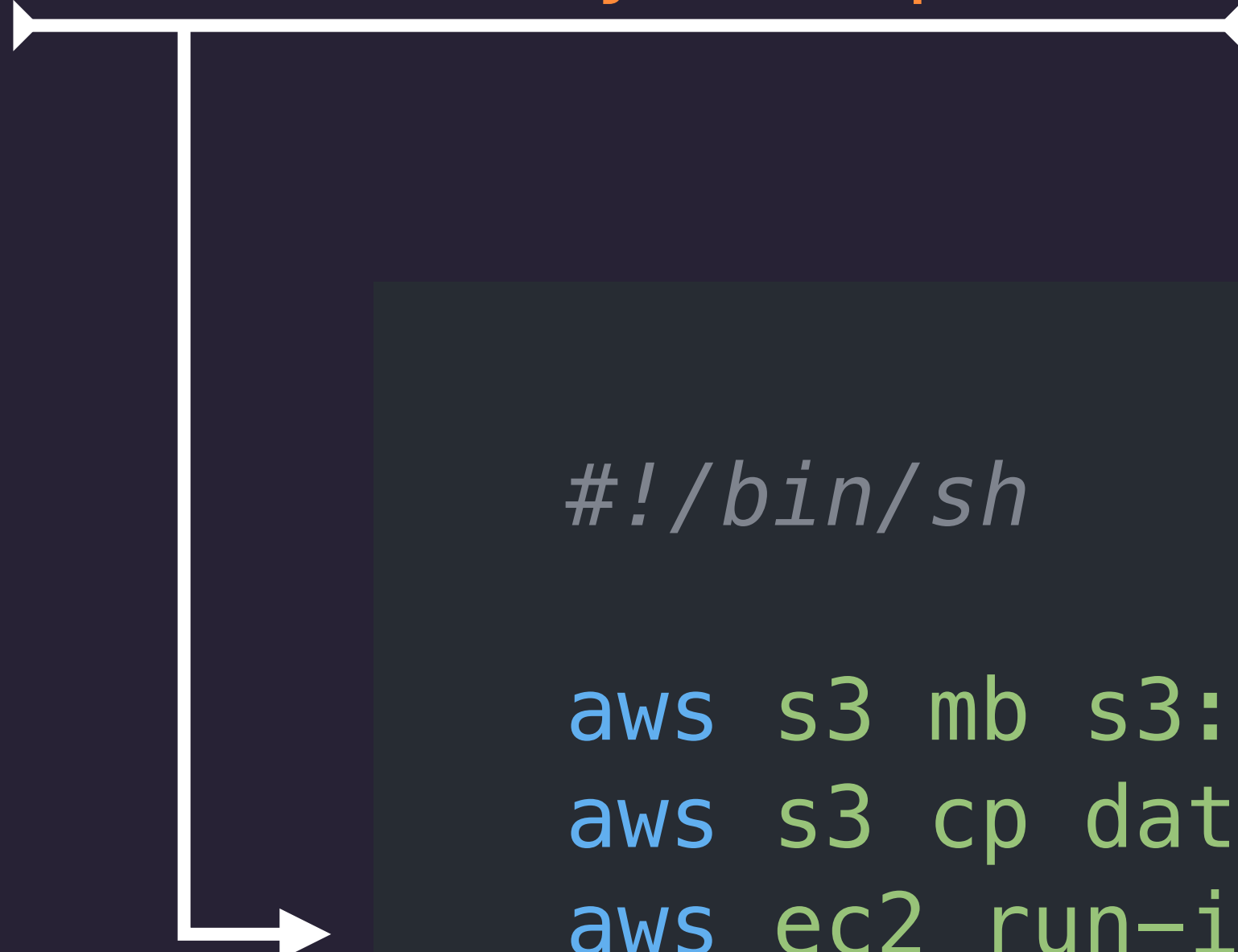
External
workflow/action



Terraform
workflow



```
resource "terraform_data" "arbitrary_script" {  
  provisioner "local-exec" {  
    working_dir = "${path.module}/scripts"  
    command     = "./arbitrary-script.sh"  
  }  
}
```

A white arrow originates from the 'command' attribute in the Terraform configuration and points to a dark grey box containing the shell script. Another white arrow originates from the 'working_dir' attribute and points to the same box, indicating that the script is executed in that directory.

```
#!/bin/sh
```

```
aws s3 mb s3://demo-bucket-12345  
aws s3 cp data.txt s3://demo-bucket-12345/  
aws ec2 run-instances \  
  --image-id ami-0123456789abcdef0 \  
  --instance-type c5.4xlarge \  
  --count 100  
aws dynamodb create-table \  
  --table-name DemoTable \  
  --attribute-definitions AttributeDefinition=DemoTableKey,KeyType=HASH \  
  --key-schema KeySchema=DemoTableKey,KeyType=HASH \  
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
```

Terraform Core Workflow

After Terraform 1.14

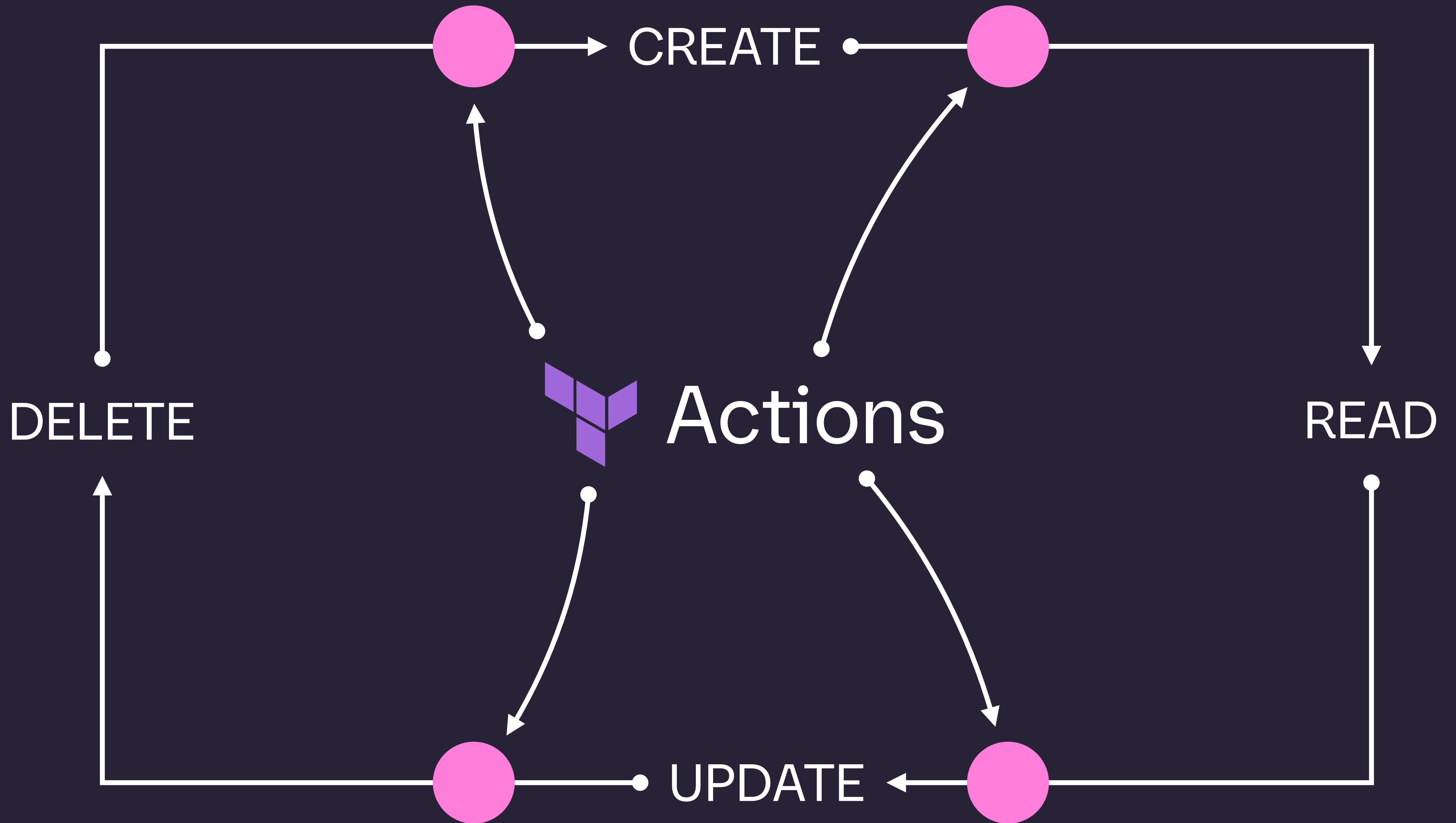
with Terraform Actions

“ Actions are preset operations built into providers that let Terraform perform day-two operations.

Terraform Actions

≠

GitHub Actions



```
action "<type>" "<name>" {  
    config {  
        # ...  
    }  
}
```

```
action "aws_lambda_invoke" "slack" {  
  config {  
    function_name = "terraform-actions-slack"  
  
    payload = jsonencode({  
      blocks = [  
        {  
          type = "section",  
          text = {  
            type = "mrkdwn",  
            text = "👋 Hello world!"  
          }  
        }  
      ]  
    })  
  }  
}
```

```
resource "aws_instance" "servers" {
  ami          = data.aws_ami.ubuntu.id
  instance_type = "t3.medium"

  lifecycle {
    action_trigger {
      actions    = [action.aws_lambda_invoke.slack]
      events     = [
        after_create,
        after_update
      ]
    }
  }
}
```