Creating and Using Custom Data Sources and Templates

Ned Bellavance

MICROSOFT MVP, CLOUD AND DATACENTER MANAGEMENT

@ned1313 www.nedinthecloud.com

Overview

Templates, templates, templates

- Improve code reuse
- Make configurations readable

Data sources tie configurations together

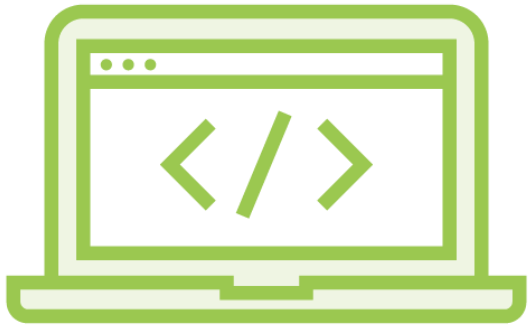
Pull configuration from external sources

Retrieve remote state information

Mo' Teams, Mo' Problems

Security Team

Wants to define policies for AWS objects



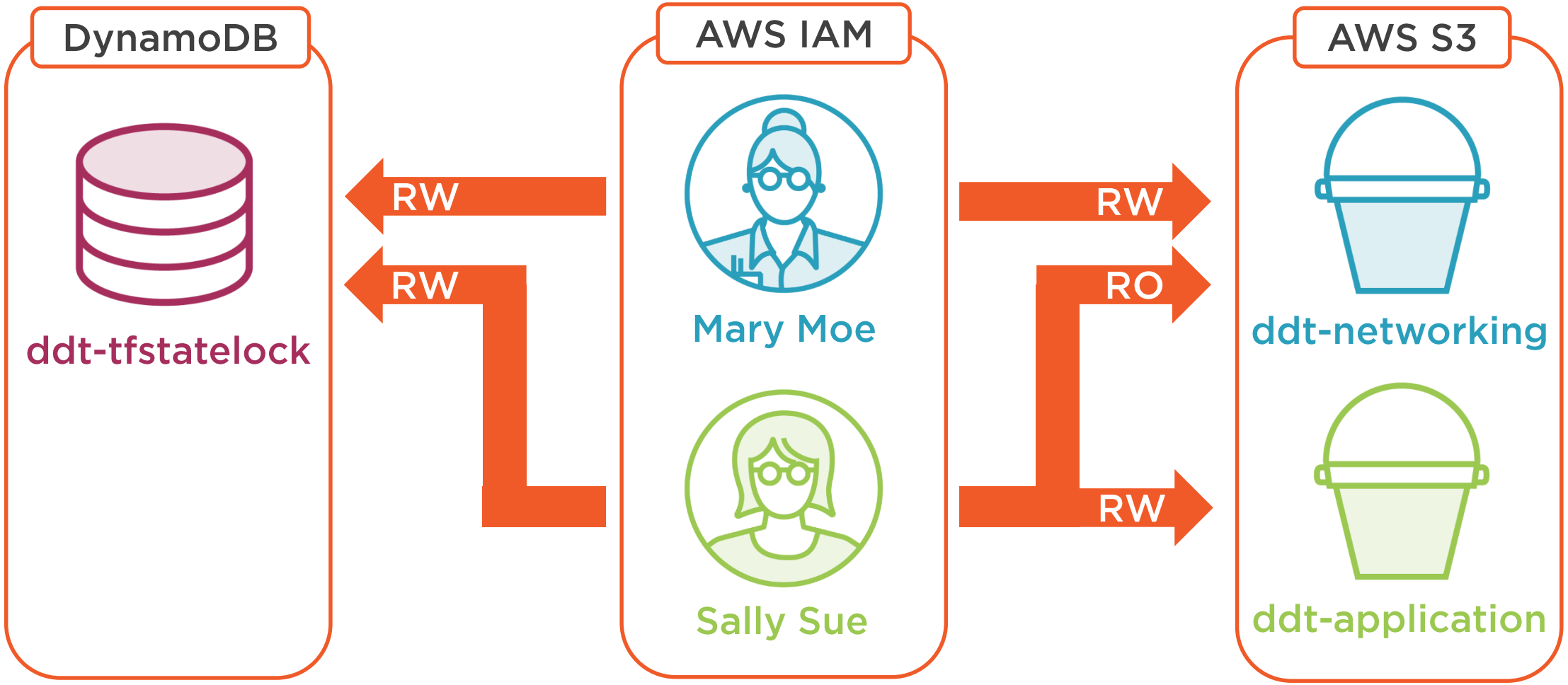
Application Team

Wants to read the network configuration and use it

Change Management Team

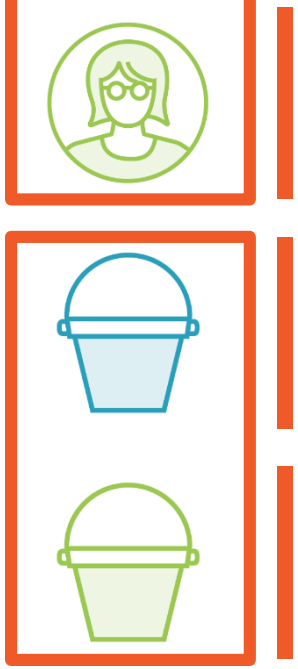
Wants all configuration settings in a repository

Remote State Setup



Roles and Policies

Read/Write access to ddt-networking, Read/Write access to ddt-statelock



Read access to ddt-networking, Read/Write access to ddt-application, Read/Write access to ddt-statelock

Allow read/write from Mary Moe, Allow read from Sally Sue

Allow read/write from Sally Sue

TEMPLATES

Long form string with interpolation

Defined by a file or in-line

Returns a rendered string only

Works with count parameter

```
data "template_file" "example" {
    count = "2"
    template = "$${var1-current_count}"
    vars {
        var1 = "${var.some_value}"
        current_count = ${count.index}
    }
}
#using template string
${data.template_file.example.rendered}
```

Example Template – In Line

Template Configuration

```
data "template_file" "s3_user_policy" {

  template =
    "${file("templates/user_policy.tpl")}"

vars {

  s3_bucket =
    "${var.aws_application_bucket}"

  }

}
```

Template File

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "s3:*",
         "Resource": [

"arn:aws:s3:::${s3_bucket}",

"arn:aws:s3:::${s3_bucket}/*"
            ]
      },
   ]
}
```

Example Template – File

Demo

Examine the Terraform setup file

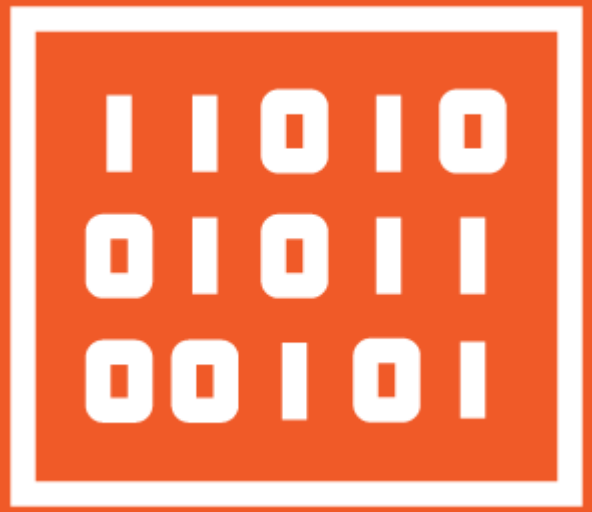
Deploy the configuration

Review the results

Play along!

- AWS account
- Terraform software (terraform.io)
- Demo files

DATA SOURCES



Data sources are the glue for multiple configurations

Resources are also data sources

Providers have specialized data sources

Alternate data sources are available

- Templates
- HTTP
- External

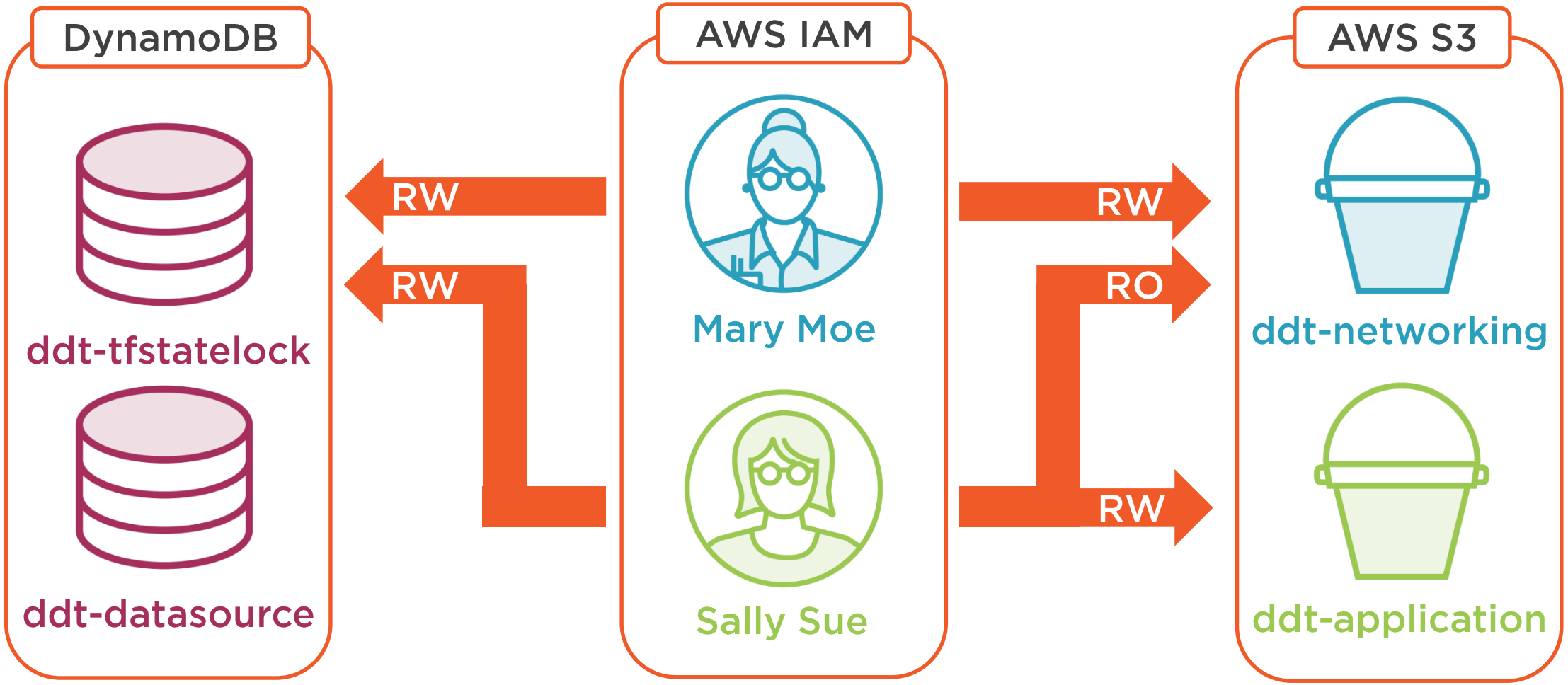
```
data "http" "example" {
    url = "https://url.to.website.com"

    request_headers  {
        header_name = "header_value"

    }
}

#using the response – treat as string
${data.http.example.body}
```

Example HTTP Data Source

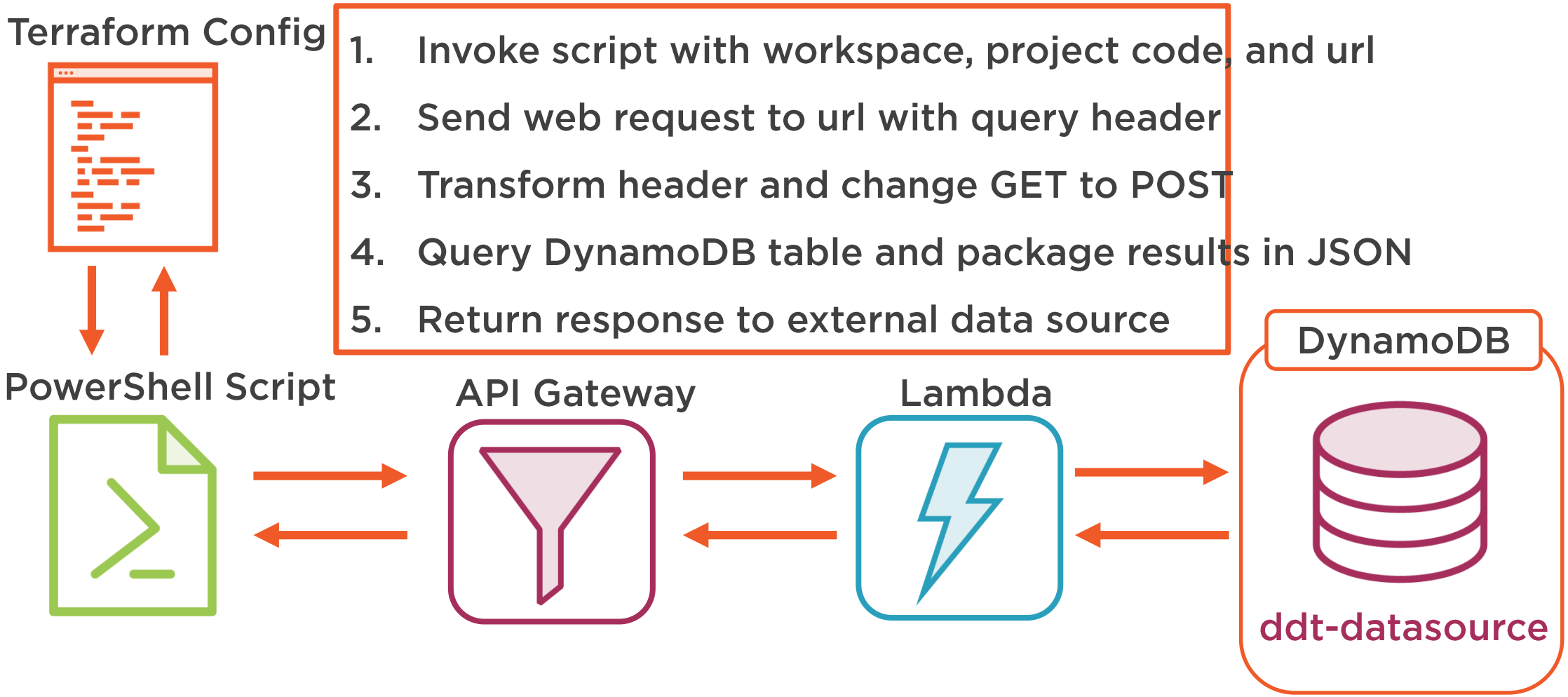
```
data "external" "example" {
    program = ["name of program", "path to script"]

    query =  {
        var1 = "value1"
        var2 = "value2"
    }
}
#using the response – treat as json
${data.external.example.result.some_value}
```

Example External Data Source

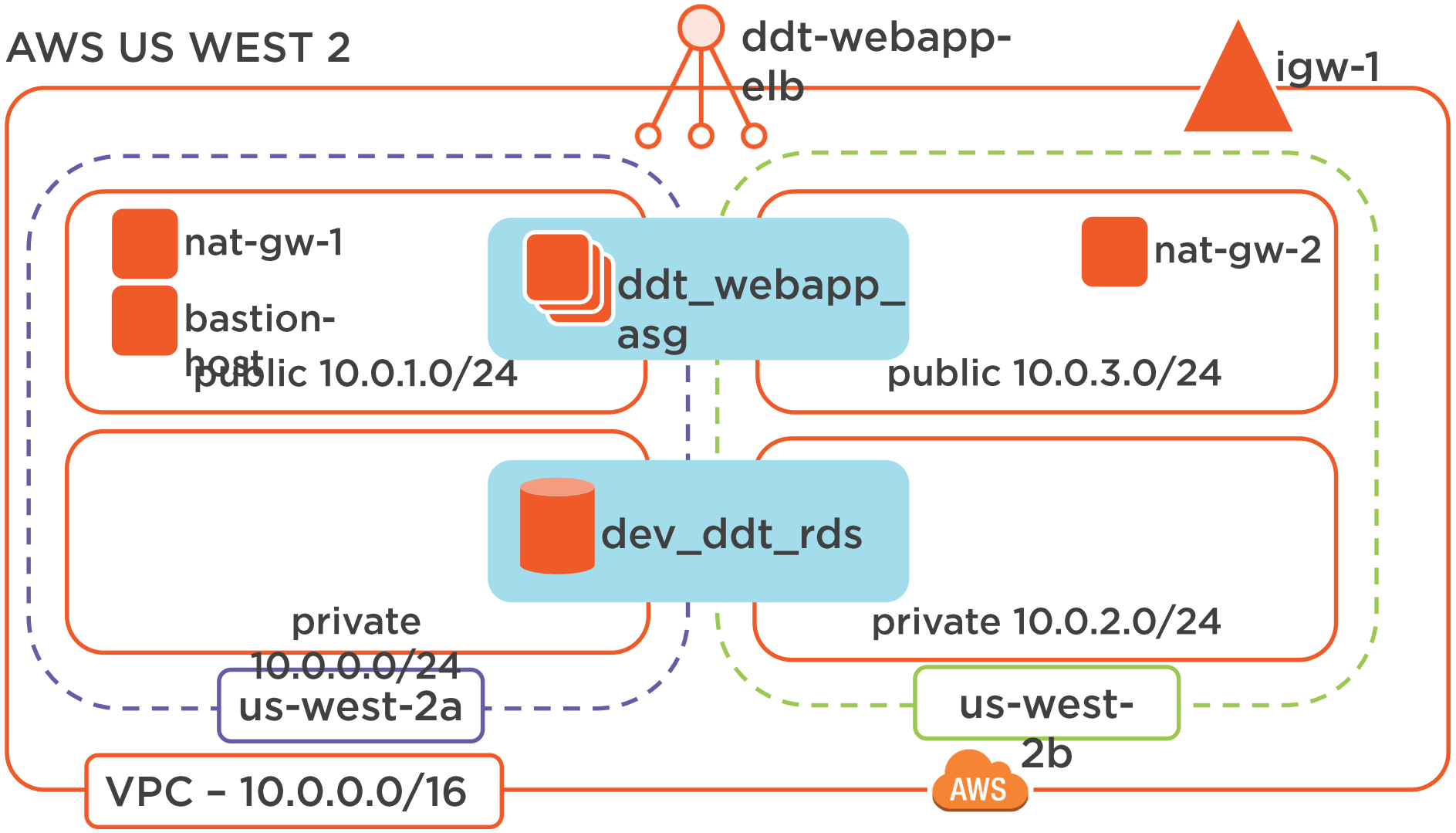
Remote State Setup



External Data Source



Application Configuration



Summary

Templates improve code reuse

Data sources glue together configurations

The only limit is your imagination

Coming up:

- Integrating Terraform with Jenkins
- Using Terraform with a CICD pipeline