

# Creating and Using Data Sources and Templates

---



**Ned Bellavance**

FOUNDER, NED IN THE CLOUD LLC

@ned1313 [www.nedinthecloud.com](http://www.nedinthecloud.com)



# Overview



**Data source types**

**Config from external sources**

**Templates, templates, templates**

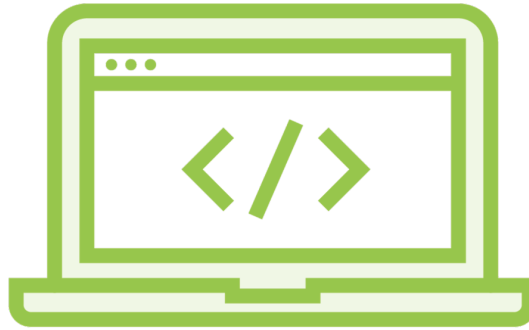


# More Teams, More Problems



## Information Security

Define roles, policies,  
and groups



## Software Development

Read network  
configuration for app  
deployment

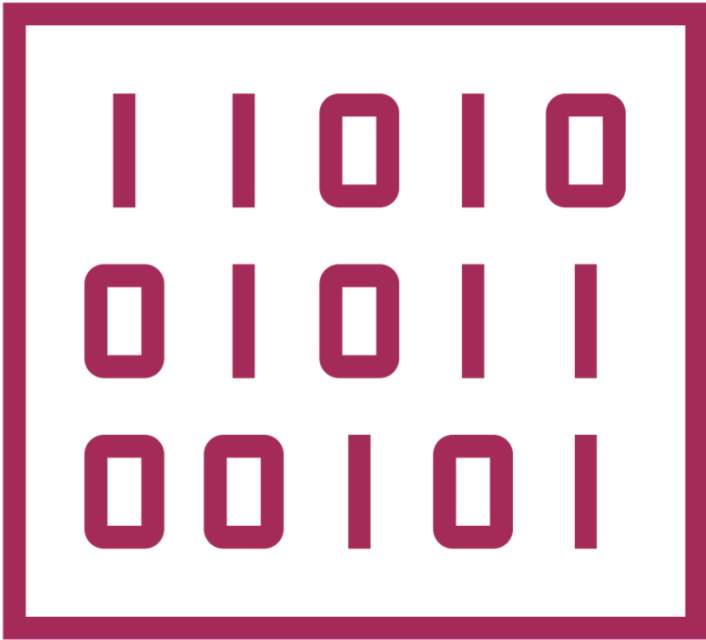


## Change Management

Store configuration  
data centrally



# Data Sources



Glue for multiple configurations

Resources are data sources

Providers have data sources

Alternate data sources

- Templates
- HTTP
- External
- Consul



# HTTP Data Source

# Example data source

```
data "http" "my_ip" {
```

```
  url = "http://ifconfig.me"
```

```
}
```

# Using the response

```
data.http.my_ip.body
```



# Consul Data Source

# Consul data source

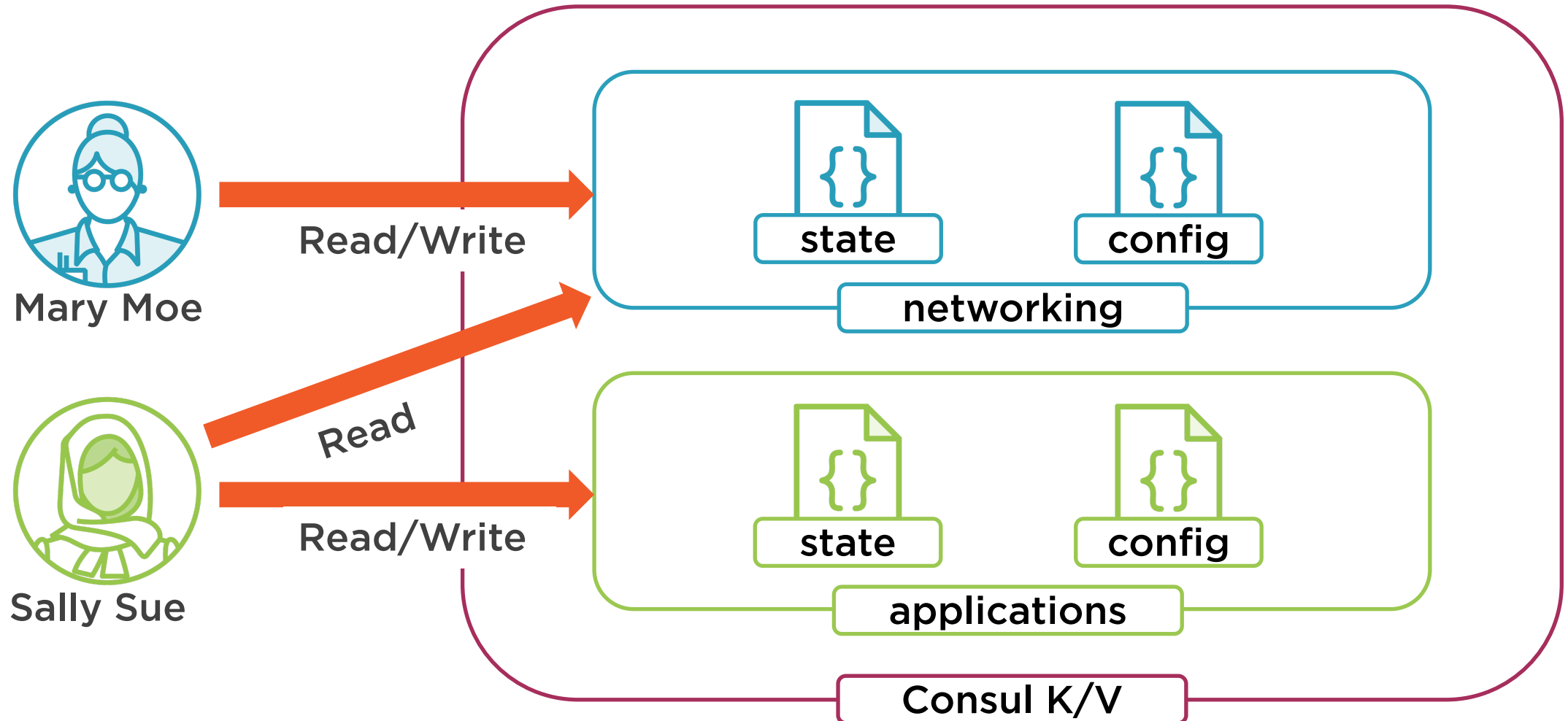
```
data "consul_keys" "networking" {  
  
  key {  
    name = "vpc_cidr_range"  
    path = "networking/config/vpc/cidr_range"  
    default = "10.0.0.0/16"  
  }  
}
```

# Using the response

```
data.consul_keys.networking.var.vpc_cidr_range
```



# Consul Setup



# Templates



## Manipulation of strings

### Overloaded term

- Quoted strings
- Heredoc syntax
- Provider
- Function

## Interpolation and directives



# Simple interpolation

```
"${var.prefix}-app"
```

# Conditional directive

```
"%{ if var.prefix != ""}${var.prefix}-app%{ else }generic-app%{ endif }"
```

# Collection directive with heredoc

```
<<EOT
```

```
%{ for name in local.names }
```

```
  ${name}-app
```

```
%{ endfor }
```

```
EOT
```

# Template strings

You've been using these implicitly

Expressed in the configuration directly

Heredoc preferred for readability



# Template Syntax In-line

# Template data source

```
data "template_file" "example" {
```

```
    count = "2"
```

```
    template = "${var1}-${current_count}"
```

```
    vars = {
```

```
        var1 = var.some_string
```

```
        current_count = count.index
```

```
    }
```

```
}
```

# Using the template

```
data.template_file.example.rendered
```



# Template configuration

```
data "template_file" "peer-role" {  
    template = file("peer_policy.txt")  
  
    vars = {  
        vpc_arn = var.vpc_arn  
    }  
}
```

# Or templatefile function

```
templatefile("peer_policy.txt",  
    { vpc_arn = var.vpc_arn } )
```

peer\_policy.txt

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ec2:AcceptVpcPeeringConnection",  
                "ec2:DescribeVpcPeeringConnections"  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "${vpc_arn}"  
            ]  
        }  
    ]  
}
```



# Summary



**Templates for code reuse**

**Data sources glue configs together**

**Custom data sources are an option**

**Coming up:**

- Workspaces and collaboration

