

Atue como um Especialista em DevSecOps e OPA (Open Policy Agent). Eu tenho um arquivo de plano do Terraform em formato JSON. Objetivo: Escreva uma política em Rego (package terraform) que negue a criação de qualquer recurso `aws_s3_bucket_public_access_block` onde `block_public_acls`, `block_public_policy`, `ignore_public_acls` ou `restrict_public_buckets` estejam definidos como `false`.

Entrada: A política deve ler a estrutura `input.resource_changes` do JSON do Terraform.

Saída: Apenas o código Rego dentro de um bloco de código, com comentários explicando a lógica.

Considere que:  
OPA version: 1.11.0

```
rego
```

```
package terraform
```

```
# Política de negação para aws_s3_bucket_public_access_block
#
# Objetivo:
# Negar a CRIAÇÃO de qualquer recurso aws_s3_bucket_public_access_block
# quando QUALQUER um dos seguintes atributos estiver definido como false:
# - block_public_acls
# - block_public_policy
# - ignore_public_acls
# - restrict_public_buckets
#
# A política lê a estrutura:
# input.resource_changes (Terraform plan em JSON)

deny[msg] {
    # Itera sobre todas as mudanças de recursos do plano
    rc := input.resource_changes[_]

    # Garante que o recurso é do tipo correto
```

```

rc.type == "aws_s3_bucket_public_access_block"

# Garante que a ação inclui criação do recurso
rc.change.actions[_] == "create"

# Obtém o estado final planejado do recurso
after := rc.change.after

# Verifica se QUALQUER um dos atributos de proteção está false
(
  after.block_public_acls == false
  or after.block_public_policy == false
  or after.ignore_public_acls == false
  or after.restrict_public_buckets == false
)

# Mensagem de erro clara para o usuário/CI
msg := sprintf(
  "Criação negada: o recurso %s (%s) possui configurações inseguras de aces
  [rc.name, rc.address]
)
}

```

 Copiar código


O código gerado apresentou erro de sintaxe ao ser validado pelo OPA. Erro: "policy.rego "data.terraform.deny"  
 1 error occurred: s3\_policy.rego:32: rego\_parse\_error: non-terminated expression  
 or after.block\_public\_policy == false  
 ^ "

rego

package terraform

```

# Política de negação para aws_s3_bucket_public_access_block
#
# Objetivo:

```

```
# Negar a CRIAÇÃO de qualquer recurso aws_s3_bucket_public_access_block
# quando QUALQUER um dos seguintes atributos estiver definido como false:
# - block_public_acls
# - block_public_policy
# - ignore_public_acls
# - restrict_public_buckets
#
# A política lê a estrutura:
# input.resource_changes (Terraform plan em JSON)
#
# Compatível com OPA 1.11.0
```

```
deny[msg] {
    # Itera sobre todas as mudanças de recursos do plano
    rc := input.resource_changes[_]

    # Filtra apenas o recurso desejado
    rc.type == "aws_s3_bucket_public_access_block"

    # Garante que a ação seja criação
    rc.change.actions[_] == "create"

    # Estado final planejado do recurso
    after := rc.change.after

    # Qualquer configuração insegura invalida o recurso
    after.block_public_acls == false
} {
    rc := input.resource_changes[_]
    rc.type == "aws_s3_bucket_public_access_block"
    rc.change.actions[_] == "create"
    after := rc.change.after
    after.block_public_policy == false
} {
    rc := input.resource_changes[_]
    rc.type == "aws_s3_bucket_public_access_block"
    rc.change.actions[_] == "create"
    after := rc.change.after
    after.ignore_public_acls == false
} {
    rc := input.resource_changes[_]
    rc.type == "aws_s3_bucket_public_access_block"
    rc.change.actions[_] == "create"
```

```
after := rc.change.after
after.restrict_public_buckets == false

msg := sprintf(
    "Criação negada: o recurso %s (%s) possui configurações inseguras de aces
    [rc.name, rc.address]
)
}
```

 Copiar código

### Observação técnica importante

No Rego (OPA 1.11.0), quebras de linha com **or** **não são permitidas** dentro de um único bloco lógico como em outras linguagens.

A forma mais segura e idiomática é usar **múltiplas regras com o mesmo nome (deny)**, onde **qualquer uma que seja satisfeita gera a negação**.



O código gerado apresentou erro de sintaxe ao ser validado pelo OPA. Erro: 8 errors occurred:  
s3\_policy.rego:18: rego\_parse\_error: if keyword is required before rule body  
s3\_policy.rego:18: rego\_parse\_error: contains keyword is required for partial set rules  
s3\_policy.rego:33: rego\_parse\_error: if keyword is required before rule body  
s3\_policy.rego:33: rego\_parse\_error: contains keyword is required for partial set rules  
s3\_policy.rego:39: rego\_parse\_error: if keyword is required before rule body  
s3\_policy.rego:39: rego\_parse\_error: contains keyword is required for partial set rules  
s3\_policy.rego:45: rego\_parse\_error: if keyword is required before rule body  
s3\_policy.rego:45: rego\_parse\_error: contains keyword is required for partial set rules

rego

```
package terraform

# Política de negação para aws_s3_bucket_public_access_block
#
# Objetivo:
# Negar a CRIAÇÃO de qualquer recurso aws_s3_bucket_public_access_block
# quando QUALQUER um dos seguintes atributos estiver definido como false:
# - block_public_acls
# - block_public_policy
# - ignore_public_acls
# - restrict_public_buckets
#
# Entrada:
# - input.resource_changes (Terraform plan em JSON)
```

☰ ☰ > ChatGPT ▾ 📺 Promoção gratuita × ✎ ⬆️ ...

```
deny[msg] if {
    # Itera sobre as mudanças de recursos do plano
    rc := input.resource_changes[_]

    # Apenas o recurso desejado
    rc.type == "aws_s3_bucket_public_access_block"

    # Apenas quando houver criação do recurso
    rc.change.actions[_] == "create"

    # Estado final planejado
    after := rc.change.after

    # Lista dos campos obrigatórios de proteção
    public_access_flags := [
        after.block_public_acls,
        after.block_public_policy,
        after.ignore_public_acls,
        after.restrict_public_buckets
    ]

    # Se QUALQUER um deles for false, a política nega
    public_access_flags[_] == false

    # Mensagem de erro retornada pelo deny
    msg := sprintf(
```

📋 Copiar código

```
        "Criação negada: o recurso %s (%s) possui configurações inseguras de aces  
        [rc.name, rc.address]  
    )  
}
```



### Por que esta versão funciona no OPA 1.11.0?

- Usa Rego v1, que exige o `if` antes do corpo da regra.
- Evita múltiplos blocos `{}` para a mesma regra `deny`, que causavam os erros de:
  - `if keyword is required before rule body`
  - `contains keyword is required for partial set rules`
- Substitui `or` por um padrão idiomático e seguro:
  - **lista + indexação** (`list[_] == false`), que expressa "qualquer elemento falso".

Essa forma é a **mais estável e recomendada** para políticas OPA em pipelines Terraform + CI/CD.

