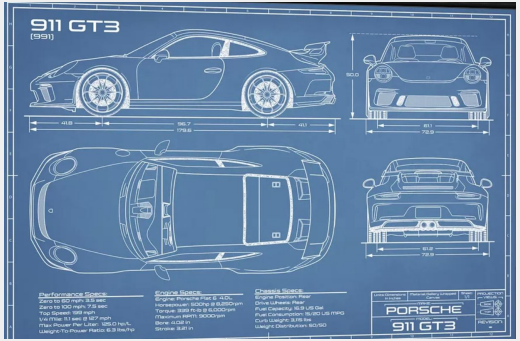
In-repo Open Policy Agent validation of GitOps definitions

Motivation of using “in-repo” OPA versus GateKeeper (blue print check versus conveyer belt check)

	
• Cheap • Agnostic of fulfillment • Validation done locally • Easy to develop • Policy validated in PR	• Expensive • Kubernetes specific • Requires k8s runtime • Requires knowledge of CR • Policy validated at fulfillment phase, post PR merge

Example use case: gitops KCC GCP project IAM definitions (roles)

Branch: master ▾ cpa-kcc-configs / project_vars / kohlsdev-cpa-inspec / iam-policy-members.yml

TKMAM6X [CPA-2505][add]added opa config and new project vars

1 contributor

19 lines (18 sloc) | 372 Bytes

```yaml
---
iamPolicyMembers:
  usersByEmail:
    mark.doll@kohls.com:
      roles:
      - viewer
    hamilton.hoover@kohls.com:
      roles:
      - editor
      - bigquery.jobUser
  groupsByEmail:
    gcp-et-devops-l3@kohls.com:
      roles:
      - viewer
  serviceAccountsByEmail:
    gitops-example-2@kohlsdev-cpa-inspec.iam.gserviceaccount.com:
      roles:
      - viewer
```

IAM roles are requested for users, groups and service accounts

Can we restrict role allocation automatically before change can be approved ?

Can we define those restriction based on environment of the project (prod vs lle) ?

Can we restrict role allocation based on requestor type: user, group, service account ?

Can we run validation automatically ?

IAM Policy definition (separate policy repository)

Branch: master ▾ cpa-kcc-templates / opa / data / iam-roles.yml

TKMAM6X added new iam roles defs

1 contributor

51 lines (50 sloc) | 1.5 KB

```yaml
---
POLICY_DATA:
  iam:
    allowedRoles:
      # PRODUCTION
      prd:
        roles:
        - editor:
            allowedForGroups: false
            allowedForServiceAccounts: true
            allowedForUsers: false
        - viewer:
            allowedForGroups: true
            allowedForServiceAccounts: true
            allowedForUsers: true
        - compute.instanceAdmin.v1:
            allowedForGroups: true
            allowedForServiceAccounts: true
            allowedForUsers: false
      # LLE
      lle:
        roles:
        - editor:
            allowedForGroups: true
            allowedForServiceAccounts: true
            allowedForUsers: true
        - viewer:
            allowedForGroups: true
            allowedForServiceAccounts: true
```

Allowed IAM roles are whitelisted per environment with additional is flags specifying availability per type of requestor

If role is not listed, it is not allowed

This content is managed independently from gitops configuration data by policy maintainers

How policies logic are defined ?

Policy is combination of policy logic (REGO file) and policy data that allows for parameterized evaluation. For example,

Policy data	REGO file
Branch: master ▾ cpa-kcc-templates / opa / data / iam-roles.yml TKMAM6X added new iam roles defs 1 contributor 51 lines (50 sloc) | 1.5 KB (see below)	Branch: master ▾ cpa-kcc-templates / opa / rego / iam-roles.rego Find file Copy path TKMAM6X [CPA-2505][add]added opa definitions 930098b 8 hours ago 1 contributor 54 lines (44 sloc) | 2.06 KB Raw Blame History (see below)

```yaml
---
POLICY_DATA:
  iam:
    allowedRoles:
      # PRODUCTION
      prd:
        roles:
        - editor:
            allowedForGroups: false
            allowedForServiceAccounts: true
            allowedForUsers: false
        - viewer:
            allowedForGroups: true
            allowedForServiceAccounts: true
            allowedForUsers: true
        - compute.instanceAdmin.v1:
            allowedForGroups: true
            allowedForServiceAccounts: true
            allowedForUsers: false
      # LLE
      lle:
        roles:
        - editor:
            allowedForGroups: true
            allowedForServiceAccounts: true
            allowedForUsers: true
        - viewer:
            allowedForGroups: true
            allowedForServiceAccounts: true
```

```
package kohls.gitops.kcc.gcp_project.iam.roles
default allow = false

environment := input.project.labels["environment-type"]

# users with associated IAM roles
users := input.iamPolicyMembers.usersByEmail
user_roles := [item | item := users[_].roles]
# flatten it
requested_user_roles := [item | item := user_roles[_][_] ]
# all allowed user IAM roles in given environment
allowed_user_roles := [role | item := data.POLICY_DATA.iam.allowedRoles[environment].roles[index][role]; data.PO

# groups with associated IAM roles
groups := input.iamPolicyMembers.groupsByEmail
group_roles := [item | item := groups[_].roles]
# flatten it
requested_group_roles := [item | item := group_roles[_][_] ]
# all allowed group IAM roles in given environment
allowed_group_roles := [role | item := data.POLICY_DATA.iam.allowedRoles[environment].roles[index][role]; data.P

# Service Accounts with associated IAM roles
sas := input.iamPolicyMembers.serviceAccountsByEmail
sa_roles := [item | item := sas[_].roles]
# flatten it
requested_sa_roles := [item | item := sa_roles[_][_] ]
# all allowed Service Account IAM roles in given environment
allowed_sa_roles := [role | item := data.POLICY_DATA.iam.allowedRoles[environment].roles[index][role]; data.POLI

# Allow if "all and each" of requested "kind" roles are present in allowed_<kind>_roles
allow {
    all_in(requested_user_roles, allowed_user_roles)
    all_in(requested_group_roles, allowed_group_roles)
    all_in(requested_sa_roles, allowed_sa_roles)
}

# ----------  Library routines to implement all_in() and in_list() validation -------------
all_in( items, list) {
    not_any_not_in(items, list)
}

not_any_not_in(items, list) {
    l = [item | item := not_in_list(items[index], list)]
    count(l) == 0
}

not_in_list(item, list) {
    not in_list(item, list)
}

in_list(item, list) {
      list[_] = item
}
```

How OPA policy is evaluated

- Policy is evaluated by combining policy logic (REGO) with policy data and “evaluating” it against context of gitops data definitions known as “input”
- Policy evaluation results in JSON output that will be interpreted as “allowed” or “not allowed”
- Policy evaluator is implemented as GitHub linter associated with repository that needs to be checked

How OPA policies validation is enabled ?

Policy is enabled by configuring **Jenkinsfile.opa** and **opa.yml** in git ops **configuration repository** to “link” gitops definitions with policy definitions located in **different repository**. Jenkins.opa file triggers opa linter and informs it about location of the policies and also provides information on how to prepare input data.

Branch: master | cpa-kcc-configs / Jenkinsfile.opa

TKMAM6X [CPA-2505][add]corrected entry point for linter

1 contributor

6 lines (4 sloc) | 208 Bytes | Raw | Blame

```
#! /usr/bin/env groovy
//Jenkinsfile to call linting tests

//calls groovy script in githook-yaml-lint repo
opaKCCGcpResources('https://github.kohls.com/EnterpriseDevOps/cpa-kcc-templates', 'opa', 'opa.yml')
```

Branch: CPA-2505-2 | cpa-kcc-configs / opa.yml

TKMAM6X [CPA-2505-2][add]added additional project to validation

1 contributor

7 lines (6 sloc) | 109 Bytes

```
---
enabled: true
projectVarFolder: project_vars
projects:
    - kohlsdev-cpa-inspec
    - kohls-sec-xpn-lle
```

Policy repository that contains policies

Folder in that policy repository that contains policies information, policy suite definitions...

Local file that allows linter to properly load “configuration” data that are being validated

Sample content of policy repo referred to from config repo (policy_suite.yml is expected to be present)

```
├── opa
│   ├── data
│   │   └── iam-roles.yml
│   ├── policy_suite.yml
│   └── rego
│       └── iam-roles.rego
```

policy_suite.yml

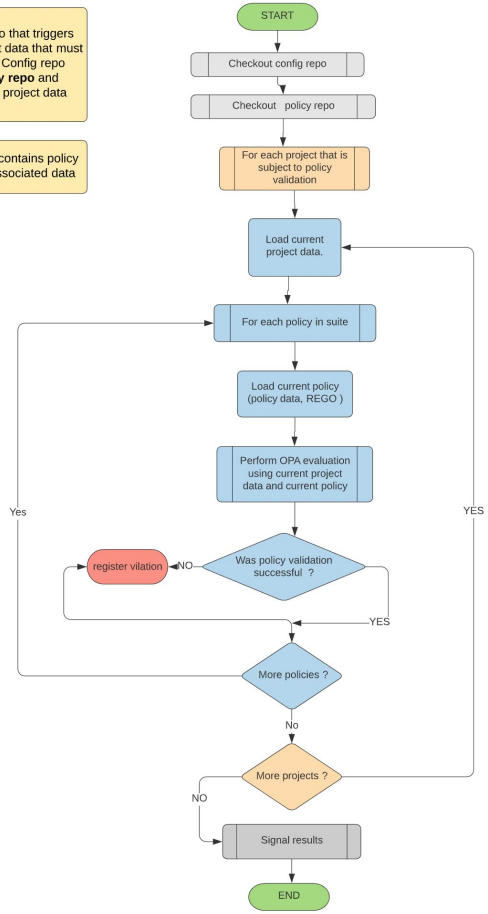
```yml
---
policies:
  - name: Users Groups and Service Accounts are only given allowed roles
    active: true
    policyDataFile: data/iam-roles.yml
    policyRegoFile: rego/iam-roles.rego
    package: "kohls.gitops.kcc.gcp_project.iam.roles"


```

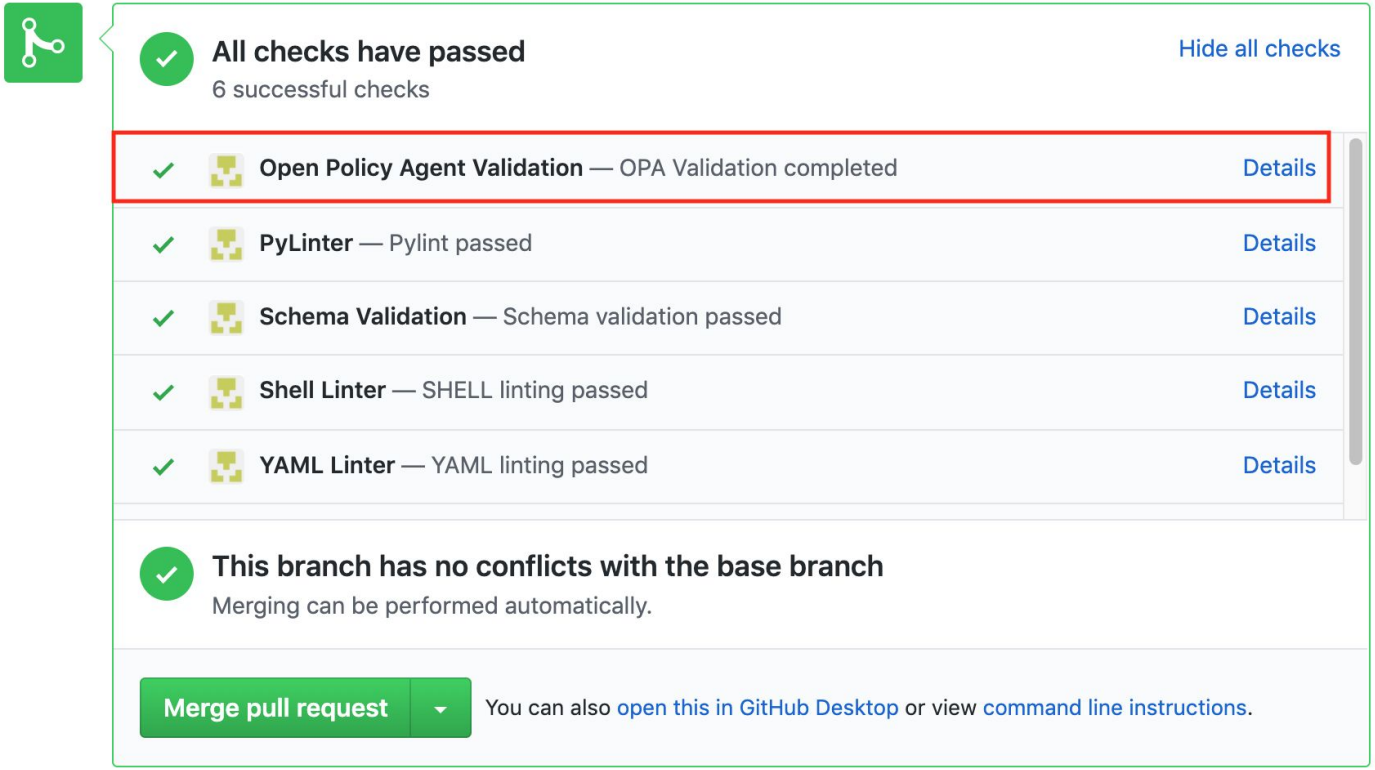
Policy validator logical workflow

Config repo is the github repo that triggers validation and it contains project data that must be checked against policies. Config repo contains reference to **policy repo** and information on how load load project data

Policy repo is the github repo contains policy suite that defines policy and associated data



Example policy validation run. GitOps view



Example policy validation run. Details (Jenkins view)

iaas2-dev1-jenkins.kohls.com:8443/blue/organizations/jenkins/githooks%2FOPA_Validation%2Fcpa-kcc-configs/detail/CPA-2505-2/3/pipeline

Apps | Kohl's Links | KOHLS APPS | myKohls | GCP console | Good Easy Reading | iaas2->paas | Training | Docs | Jenkins | GitHubRepos

- opa_validation/opa_linter.py — Load a resource file from a shared library
- opa_validation/requirements.txt — Load a resource file from a shared library
- opa_validation/opa_run.py — Load a resource file from a shared library
- requirements.txt — Write file to workspace
- main_script.py — Write file to workspace
- runner_script.py — Write file to workspace
- pip3 install -r requirements.txt --user --proxy=http://proxy-gcp-central.kohls.com:8080 — Shell Script
- Shell Script
- cat /tmp/opa.log — Shell Script

```
+ cat /tmp/opa.log
2020-07-14 16:22:28,855  INFO: Loaded OPA validation config
2020-07-14 16:22:28,856  INFO: Processing project data for kohls-sec-xpn-lle
2020-07-14 16:22:29,042  INFO: Processing policy 'Users Groups and Service Accounts are only given allowed roles'
2020-07-14 16:22:31,476  INFO: OPA evaluation compeleted
2020-07-14 16:22:31,476  INFO: Result set. START
2020-07-14 16:22:31,476  INFO: {
  "result": [
    {
      "expressions": [
        {
          "value": {
            "allow": true,
            "allowed_group_roles": [
              "editor",
              "viewer"
            ],
            "allowed_sa_roles": [
              "editor",
              "viewer",
              "bigquery.jobUser",
              "compute.networkUser",
              "compute.networkUser",
              "container.hostServiceAgentUser",
              "compute.networkUser"
            ],
            "allowed_user_roles": [
              "editor",
              "viewer",
              "bigquery.jobUser"
            ],
            "environment": "lle",
```

Interpretation of OPA policy validation response

```
2020-07-14 16:22:28,855  INFO: Loaded OPA validation config
2020-07-14 16:22:28,856  INFO: Processing project data for kohls-sec-xpn-lle
2020-07-14 16:22:29,042  INFO: Processing policy 'Users Groups and Service Accounts are only given allowed roles'
2020-07-14 16:22:31,476  INFO: OPA evaluation compeleted
2020-07-14 16:22:31,476  INFO: Result set. START
2020-07-14 16:22:31,476  INFO: {
  "result": [
    {
      "expressions": [
        {
          "value": {
            "allow": true,
            "allowed_group_roles": [
              "editor",
              "viewer"
            ],
            "allowed_sa_roles": [
              "editor",
              "viewer",
              "bigquery.jobUser",
              "compute.networkUser",
              "compute.networkUser",
              "container.hostServiceAgentUser",
              "compute.networkUser"
            ],
            "allowed_user_roles": [
              "editor",
              "viewer",
              "bigquery.jobUser"
            ],
            "environment": "lle",
            "group_roles": [],
            "requested_group_roles": [],
            "requested_sa_roles": [
              "container.hostServiceAgentUser",
              "compute.networkUser",
              "compute.networkUser",
              "compute.networkUser"
            ],
            "requested_user_roles": [],
            "sa_roles": [
              [
                "container.hostServiceAgentUser",
                "compute.networkUser"
              ],
              [
                "compute.networkUser"
              ],
              [
                "compute.networkUser"
              ]
            ],
            "sas": {
              "938246310506-compute@developer.gserviceaccount.com": {
                "roles": [
                  "compute.networkUser"
                ]
              },
              "938246310506@cloudservices.gserviceaccount.com": {
                "roles": [
                  "compute.networkUser"
                ]
              },
              "service-938246310506@container-engine-robot.iam.gserviceaccount.com": {
                "roles": [
                  "container.hostServiceAgentUser",
                  "compute.networkUser"
                ]
              }
            },
            "user_roles": []
          },
          "text": "data.kohls.gitops.kcc.gcp_project.iam.roles",
          "location": {
            "row": 1,
            "col": 1
          }
        }
      ]
    }
  ]
}

2020-07-14 16:22:31,476  INFO: Result set. END
2020-07-14 16:22:31,476  INFO: >>>>>>>>> 'Users Groups and Service Accounts are only given allowed roles' ALLOWED
```

Indicates what data are used for input and what policy is used

"allow: true"
Means policy is allowed.

Debug info: shows roles (user, group, SA) allowed by policy definition

Debug info: shows roles (user, group, SA) requested by gitops config

References

Config repo (data that are being validated)
https://github.kohls.com/EnterpriseDevOps/cpa-kcc-configs/tree/CPA-2505-2

Repo that contains policy definitions
https://github.kohls.com/EnterpriseDevOps/cpa-kcc-templates

Validator (linter) code
https://github.kohls.com/EnterpriseDevOps/linter_dev

Workflow diagram (policy validation process)
https://app.lucidchart.com/documents/view/48648ba7-481c-4833-9fb4-bb1be181e03c
https://www.openpolicyagent.org/