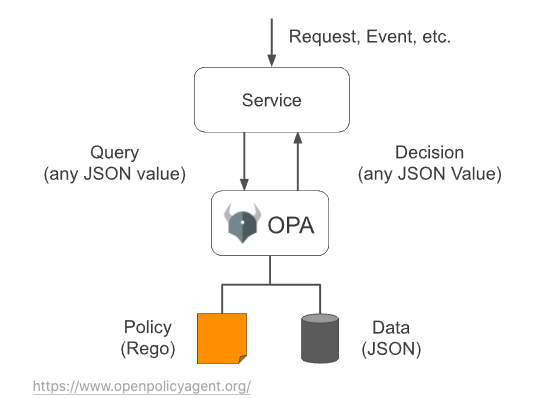
OPA?

OPA는 정책을 결정하는 부분과 수행하는 영역을 분리하기 위해 시작되었고, 전체 시스템에 걸쳐 통합된 상황에 따른 정책 시행을 가능하게하는 오픈소스 범용 정책 엔진이 되었다. OPA는 현재 CNCF의 졸업 프로젝트(Graduated project)로 클라우드 네이티브 정책의 사실상 표준으로 자리 잡고 있다. 실제로 넷플릭스, 핀터레스트 등 쿠버네티스 및 마이크로 서비스에 대한 정책 엔진으로 사용되고 있으며 아틀라시안 제품군이나 Chef에도 포함되어있다.



OPA는 쿼리 입력을 평가하고 정책 및 데이터에 대해 정책 결정을 생성한다. OPA 및 Rego는 도메인에 구애받지 않으므로 정책에서 거의 모든 종류의 불변을 설명할 수 있다.

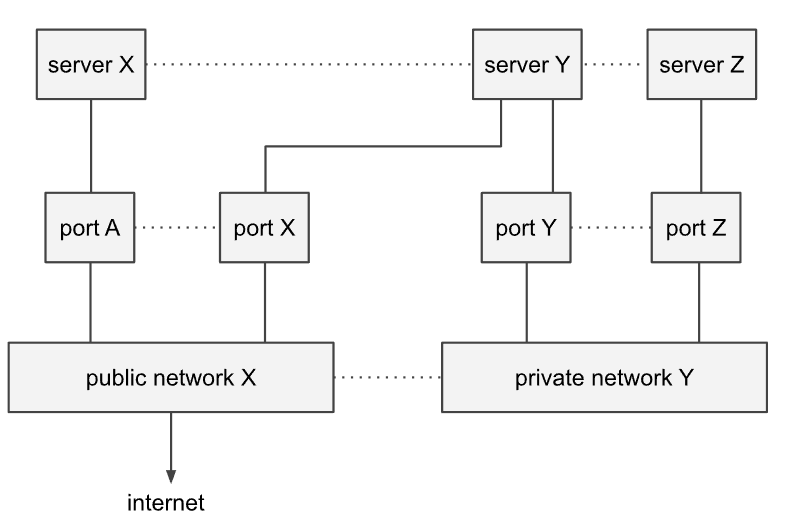
예를 들면:

* 어떤 사용자가 어떤 리소스에 액세스 할 수 있는지.
* 어떤 Subnet의 Egress Traffic이 허용되는지.
* 워크로드가 어느 클러스터에 배포되어야 하는지.
* 어떤 Registries binaries로부터 다운 받을 수 있는지.
* 언제 시스템에 접근 할 수 있는지.

**Rego**

OPA 정책은 Rego라는 고급 선운 언어로 표현된다. Rego는 복잡한 계층적 데이터 구조에 대한 정책을 표현하기 위해 특별히 제작되었다.

https://www.openpolicyagent.org/docs/latest/policy-language/



시스템에는 3가지 종류의 구성 요소가 있다.

* 서버는 0갱 이상의 프로토콜 (Ex: http, ssh등)을 노출한다.
* 네트워크는 서버를 연결하며 Public 또는 Private이 될 수 있다. Public 네트워크는 인터넷에 연결된다.
* 포드는 서버를 네트워크에 연결한다.

이를 더 구체적으로 정의 해보자.

텍스트, 실내, 검은색이(가) 표시된 사진

자동 생성된 설명위 시스템을 json으로 표현하면 아래와 같이 설명이 가능하다. 그리고 이후 설명에서 입력 데이터로 사용할 것이다. 즉, 아래의 내용으로 input.json 이라는 파일로 생성해준다.

|  |
| --- |
| {  "servers": [  {"id": "app", "protocols": ["https", "ssh"], "ports": ["p1", "p2", "p3"]},  {"id": "db", "protocols": ["mysql"], "ports": ["p3"]},  {"id": "cache", "protocols": ["memcache"], "ports": ["p3"]},  {"id": "ci", "protocols": ["http"], "ports": ["p1", "p2"]},  {"id": "busybox", "protocols": ["telnet"], "ports": ["p1"]}  ],  "networks": [  {"id": "net1", "public": false},  {"id": "net2", "public": false},  {"id": "net3", "public": true},  {"id": "net4", "public": true}  ],  "ports": [  {"id": "p1", "network": "net1"},  {"id": "p2", "network": "net3"},  {"id": "p3", "network": "net2"}  ]  } |

아래와 같은 요구사항을 충족하여 정책을 생성해주자.

1. 인터넷에서 접속 가능한 서버들은 http를 노출해서는 안됨
2. 모든 서버들은 telnet을 노출해서는 안됨

이를 rego를 사용해 정의하면 다음과 같다. 라인별 내용은 주석을 통해 확인

|  |
| --- |
| package example  default allow = false # 기본적으로 허용되지 않는다.  allow = true { # 위반이 하나도 없다면 허용한다.  count(violation) == 0  }  violation[server.id] { # 서버는 다음의 경우에 위반  some server # 모든 서버를 대상으로  public\_server[server] # public\_server 함수가 를 통해 설정되어 있거나  server.protocols[\_] == "http" # http 프로토콜을 포함하고 있다면  }  violation[server.id] { # 서버는 다음의 경우에 위반 (앞의 룰과 or조건)  server := input.servers[\_] # input에 있는 모든 서버들에 대해  server.protocols[\_] == "telnet" # telnet 프로토콜을 포함하고 있다면  }  public\_server[server] { # 서버가 public에 연결되었는지 체크  some i, j  server := input.servers[\_] # input에 있는 모든 서버를 대상으로  server.ports[\_] == input.ports[i].id # 모든 포트들의 id를 기준으로  input.ports[i].network == input.networks[j].id # 그 포트의 네트워크를 찾아서  input.networks[j].public # 그 네트워크가 public 이면  } |

그리고 위 내용으로 policy.rego 이라는 파일을 생성해준다.

OPA를 설치해주자.

|  |
| --- |
| $ curl -L -o opa https://github.com/open-policy-agent/opa/releases/download/v0.11.0/opa\_linux\_amd64  $ chmod 755 ./opa  $ mv ./opa /usr/local/bin/ |

이제 아까 만든 데이터와 정책을 집어 넣어 진행내역을 확인해주자.

|  |
| --- |
| $ opa eval -i input.json -d policy.rego "data.example.violation[x]" |

텍스트이(가) 표시된 사진

자동 생성된 설명

또 대화형식으로 정책 로딩을 할 수 있다.

|  |
| --- |
| $ opa run policy.rego repl.input:input.json  > data.example.public\_server[s] |

