

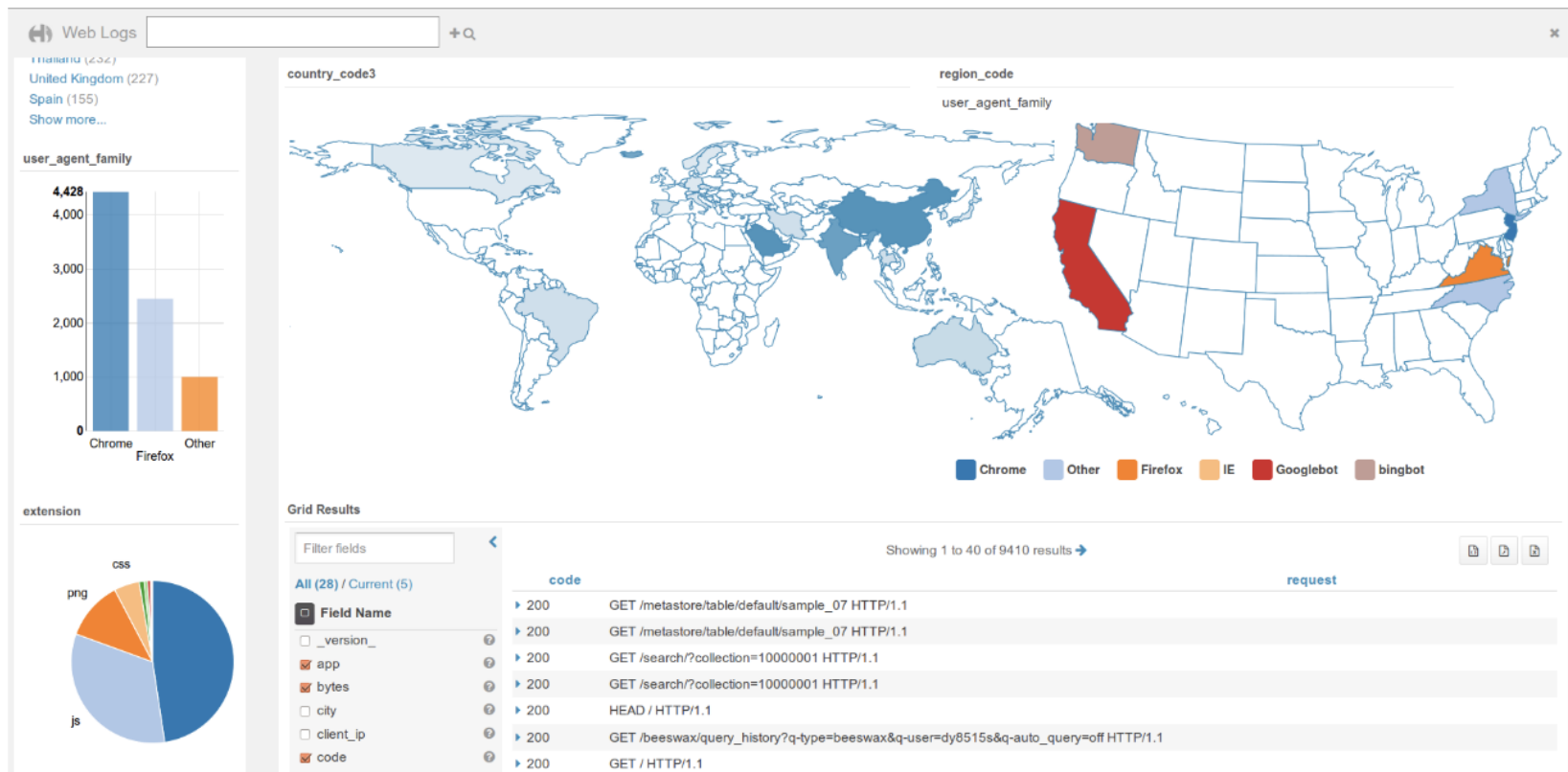
6. Using Kubernetes Resources

Hue Editor

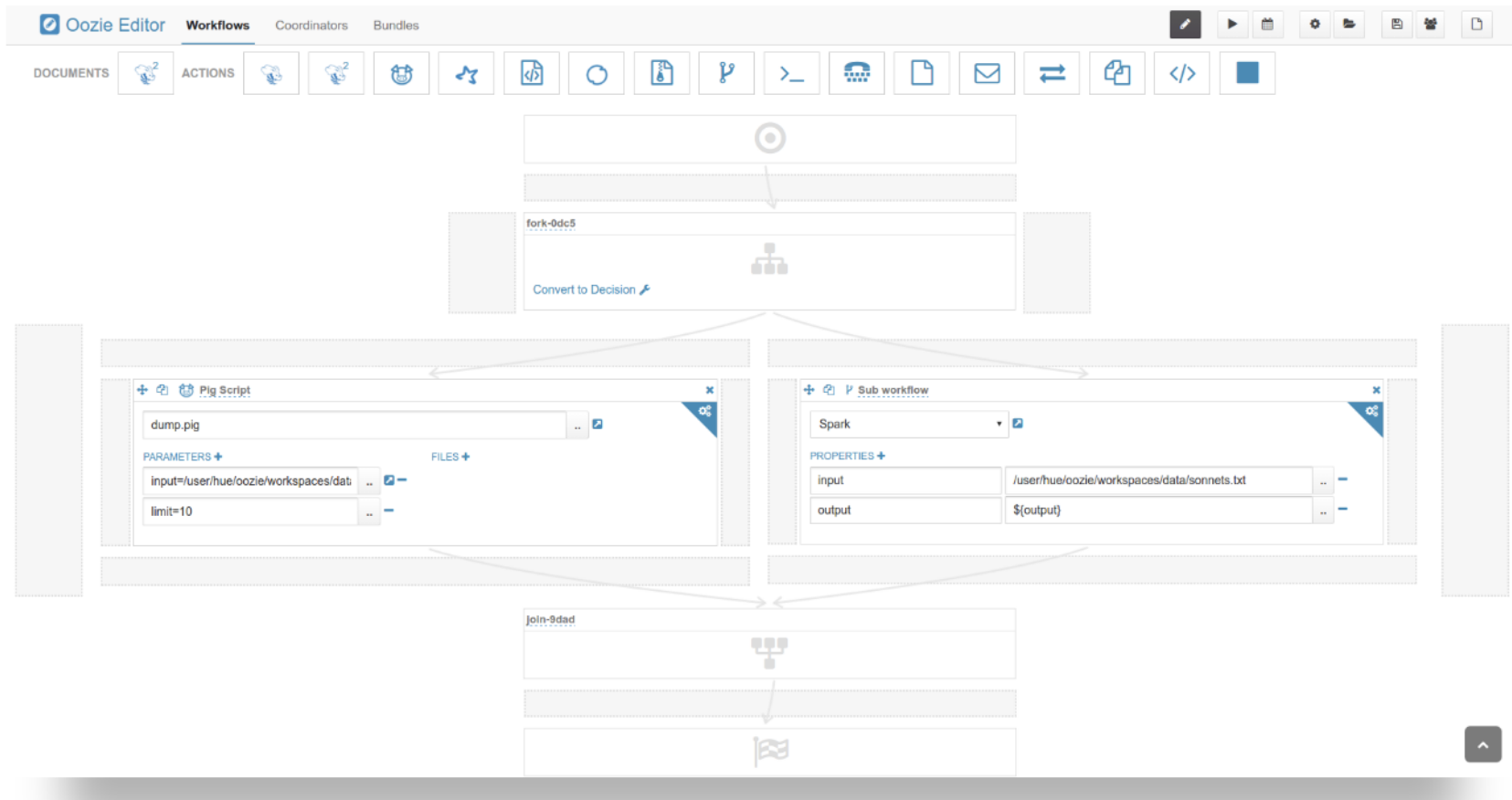
The screenshot displays the Hue Editor interface, which is used for managing Hive and Impala queries. The interface is divided into several sections:

- Top Bar:** Contains the Hue logo, a "Query" dropdown, a search bar for data and saved documents, and user information (Jobs, a refresh icon, and the user "romain").
- Left Sidebar:** Lists available tables and their schemas. Under "u_omain", tables like "account_stnew", "brinetest1", "doc_feedback_aug", "free_logs", "free_logs_2016_06", "hue_commits", "jira_c", "marktable", "query_impala", "queryresult", "queryresultfromsfcd", "sfcdqueryresult", "tes_ux", "test", "testresult", "ticket", "ticket2", "ticket3", and "ticket4" are listed.
- Central Panel:**
 - Query Editor:** Displays a SQL query with line numbers 57 to 75. The query is an Impala SELECT statement that filters for tickets with multiple SFDC tickets linked, based on JIRA names and ticket status. A warning "Some high risks were detected." is shown at the top.
 - Filters:** Below the query editor, there are input fields for "component" (set to "Hue"), "type" (set to "Escalation"), and "date" (set to "2016-10-01").
 - Query History:** A list of recent queries with their IDs and completion status.
 - Results:** A table showing the results of the query, with columns "name", "jira_summary_c", and "tickets".
- Right Sidebar:**
 - Tables:** Lists the tables used in the query: "jira_c", "cases", and "ticket".
 - Suggestions:** Provides advice on query optimization, such as "Query on partitioned table is missing filters on partitioning columns. Rewrite query to add filtering conditions." and a link to "Improve Analysis".

Hue Dashboard



Hue Workflow



Hue Browsers

The screenshot displays the Hue Table Browser interface. The top navigation bar includes the Hue logo, a 'Query' dropdown, a search bar for data and documents, and a 'Jobs 2' status indicator. The left sidebar shows a tree view of databases, with 'default' selected and 'web_logs' highlighted. The main panel shows the 'web_logs' table details, including its properties, statistics, and a list of 29 columns. A sample of the data is shown at the bottom.

Databases > default > web_logs

No description available

Overview Columns (29) Partitions (2) Sample Details

PROPERTIES

STATS

TAGS

Table
hue
01/06/2017 10:32
text Not compressed

COLUMNS (29)

	Name	Type	Comment
1	<i>i</i> _version_	bigint	Add a comment...
2	<i>i</i> app	string	Add a comment...
3	<i>i</i> bytes	smallint	Add a comment...
4	<i>i</i> city	string	Add a comment...
5	<i>i</i> client_ip	string	Add a comment...

[View more...](#)

SAMPLE

	web_logs._version_	web_logs.app	web_logs.bytes	web_logs.city	web_logs.client_ip	web_logs.code	web_logs.country_code	web_logs.country_code3	web_logs.country_name	w
1	148089575574446000	hbase	595	Hyderabad	49.206.186.56	NULL	IN	IND	India	0
2	148089575574446000	static	831	Hyderabad	49.206.186.56	NULL	IN	IND	India	0
3	148089575574446000	static	594	Hyderabad	49.206.186.56	NULL	IN	IND	India	0

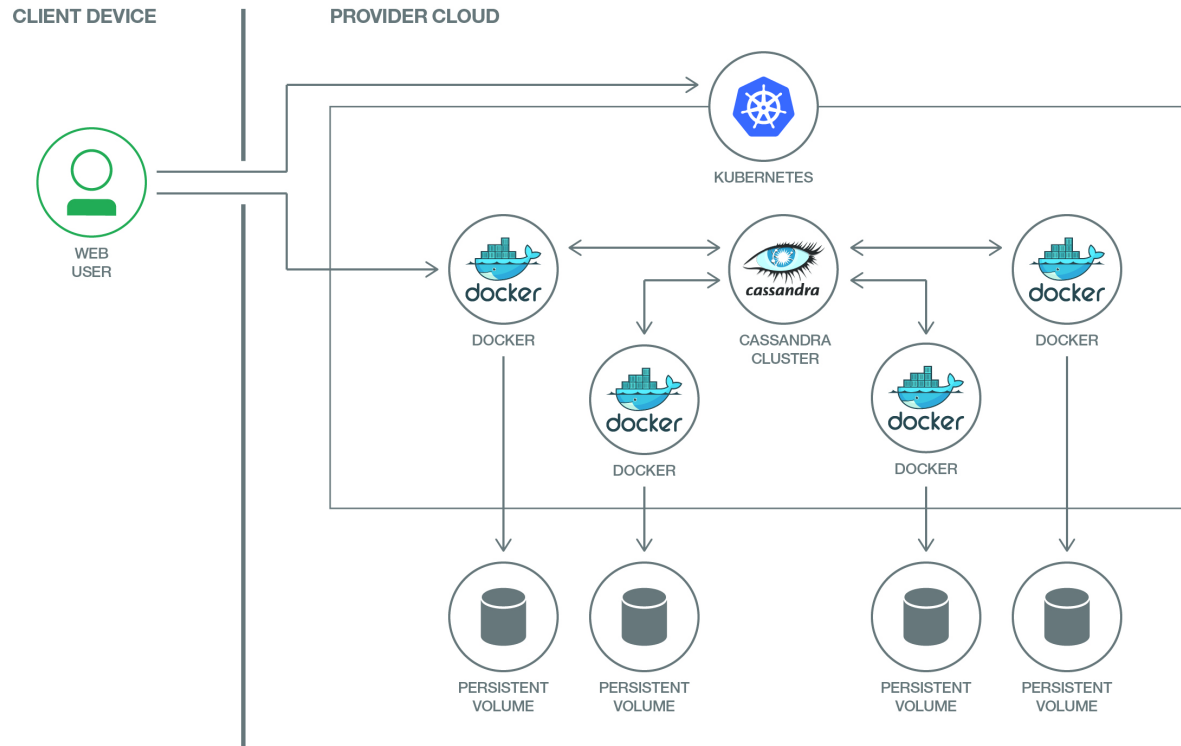
ConfigMap

- 환경 변수나 설정 값들을 변수로 관리
- Pod가 생성될 때 값들을 넣어줄 수 있음 (key-value 형태)
- DEV, PROD ConfigMap을 미리 만들어놓고 배포할 때 반영
- 값을 환경변수로 넘기거나, 디스크 볼륨으로 마운트 가능
- <https://kubernetes.io/docs/tutorials/configuration/configure-redis-using-configmap/>

Stateless vs StatefulSet

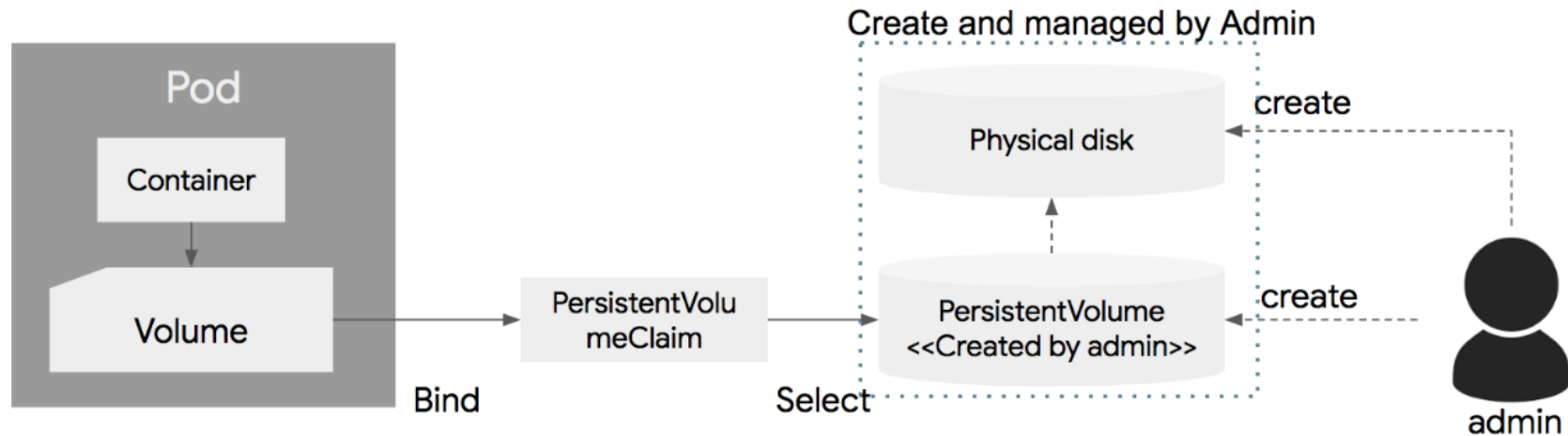
- Stateless: 컨테이너가 상태를 가지지 않는 경우
- ReplicationSet, Deployment: 상태가 없는 Pod를 관리하기 위한 용도
- 컨테이너가 죽었을 때, 데이터가 사라짐
- 상태가 없기 때문에 장애 발생 시 새로 띄우거나 옮기는게 자유로움
- 하지만 컨테이너가 내려가거나 재시작하더라도 상태가 유지되어야 하는 경우가 있음 (ex. 데이터베이스)

Stateless vs StatefulSet



- StatefulSet: 상태를 가지고 있는 Pod를 관리하는 컨트롤러 (1.9v)
- Multi-Node Cassandra, MongoDB, Zookeeper와 같은 경우
- 데이터를 보존할 수 있어야 함 (PersistentVolume 사용)

PersistentVolume, PersistentVolumeClaim

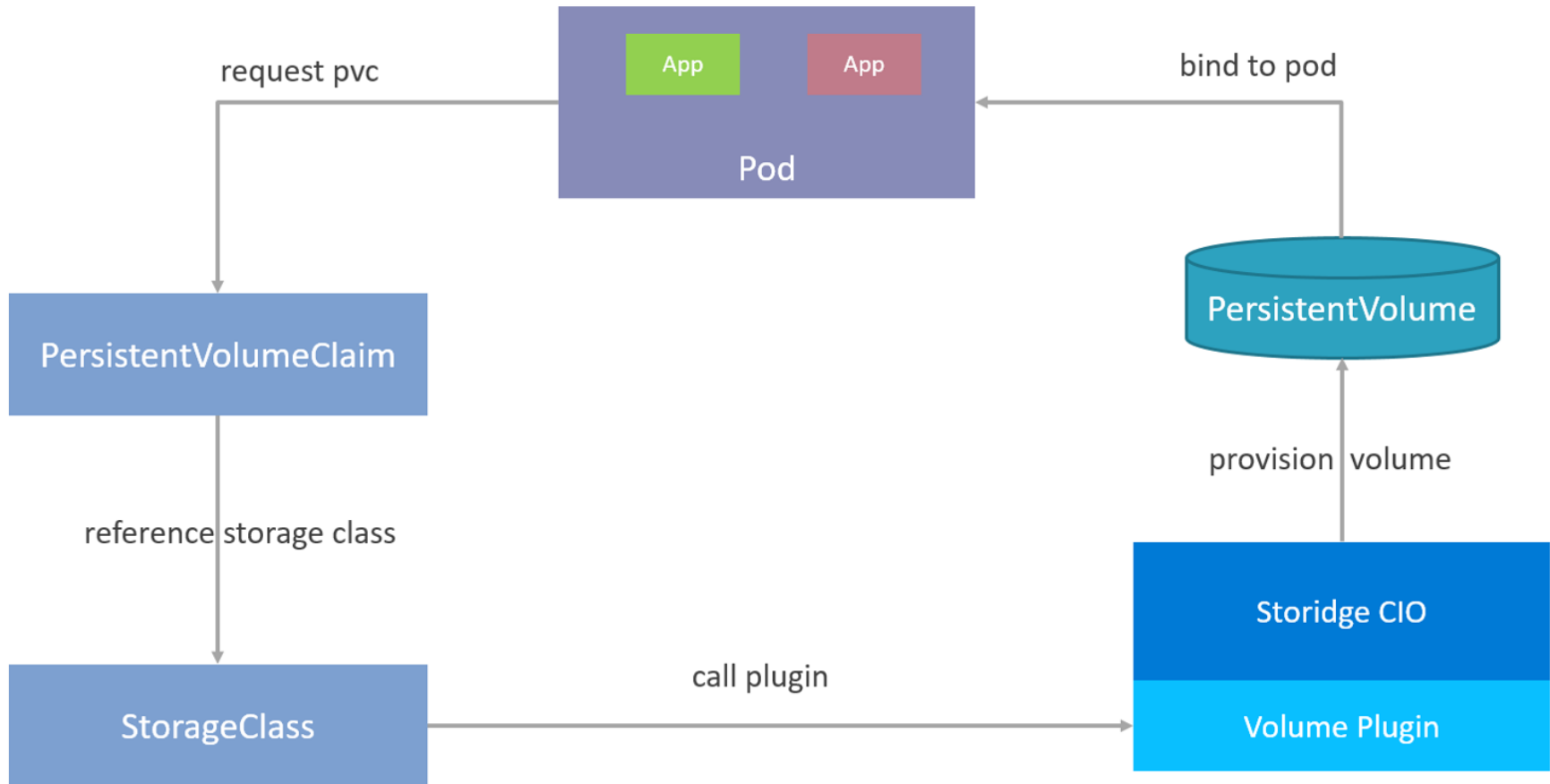


- PV는 물리적 디스크 그 자체, 클러스터 리소스로 취급
- PVC는 사용자가 PV에 하는 요청
- 왜 분리시켰을까? 인프라 관리자와 개발자의 역할을 나눔 (인프라 추상화)
- 볼륨을 Pod에 직접 할당하기보다 Pod가 사용할 스토리지를 분리

PersistentVolume, PersistentVolumeClaim

- Pod의 생성, 삭제와 상관없이 별도의 생명주기를 가짐
- Provisioning - Binding - Using - Reclaiming
- Provisioning: PV를 생성하는 단계 (Dynamic / Static)
- Binding: 만들어진 PV를 PVC와 바인딩하는 단계 (PV 할당)
- Using: PVC가 Pod에 설정된 상태, Pod가 사용 중인 PVC는 시스템에서 삭제 불가능 (Storage in Use Protection)
- Reclaiming: 사용이 끝난 PVC 삭제, 사용 중이던 PV를 초기화
- 1.9 버전부터 `VolumeMode` 지정 가능 (raw, filesystem)
- 1.11 버전부터 PVC 확장 가능 (`allowVolumeExpansion: true`)

Dynamic Provisioning



- KOPS의 경우 storage-aws addon으로 EBS StorageClass 생성
- AWS에서 Dynamic Provisioning의 경우 계속 EBS가 생성됨(?)

Tutorial

<https://github.com/Swallow/KubeStudy/tree/master/config>