Migrating Datastores

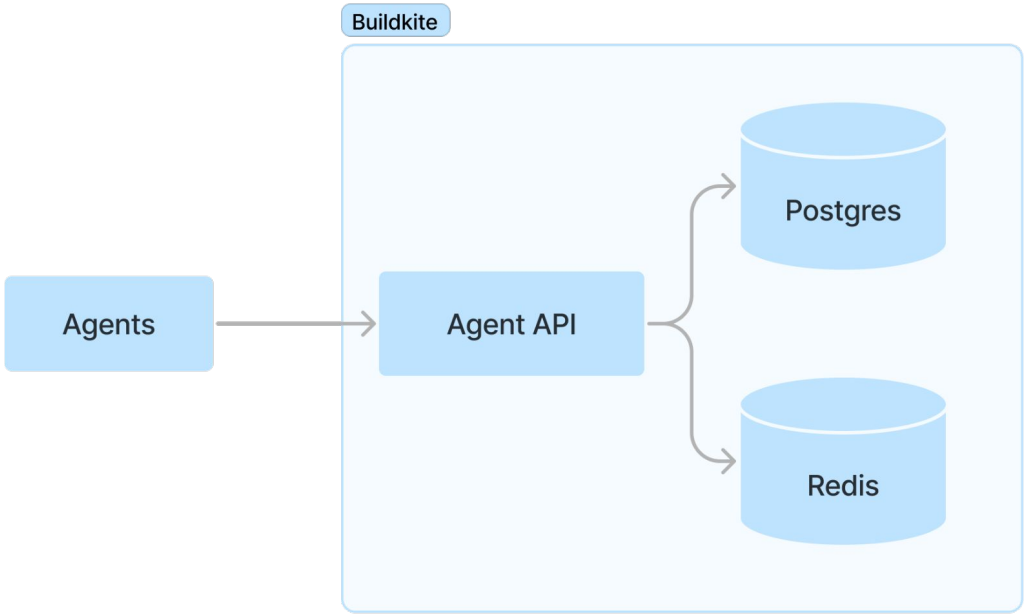
Tessa Bradbury

Who are we and what are we doing?

Buildkite

Our Team

Architecture: The interesting bits



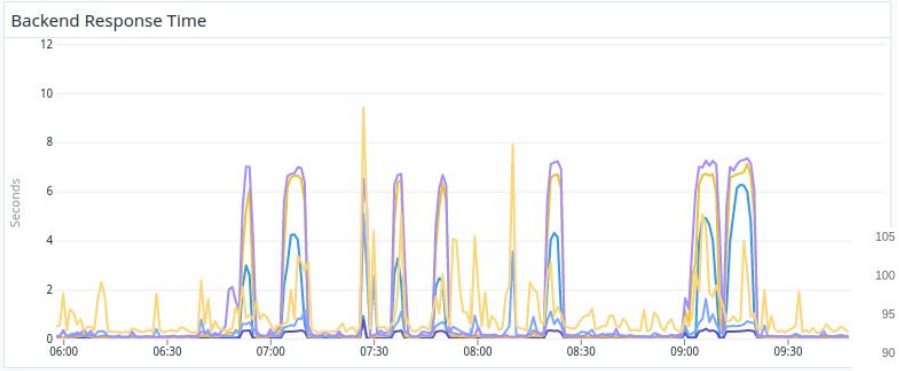
What's Redis?

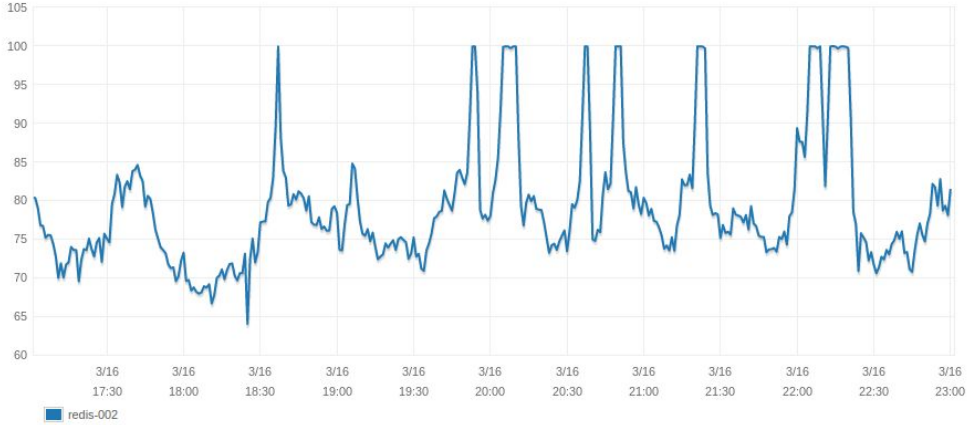
- In memory, key-value datastore
- Fast but ephemeral - if the server reboots, the data is gone

We use Redis for:

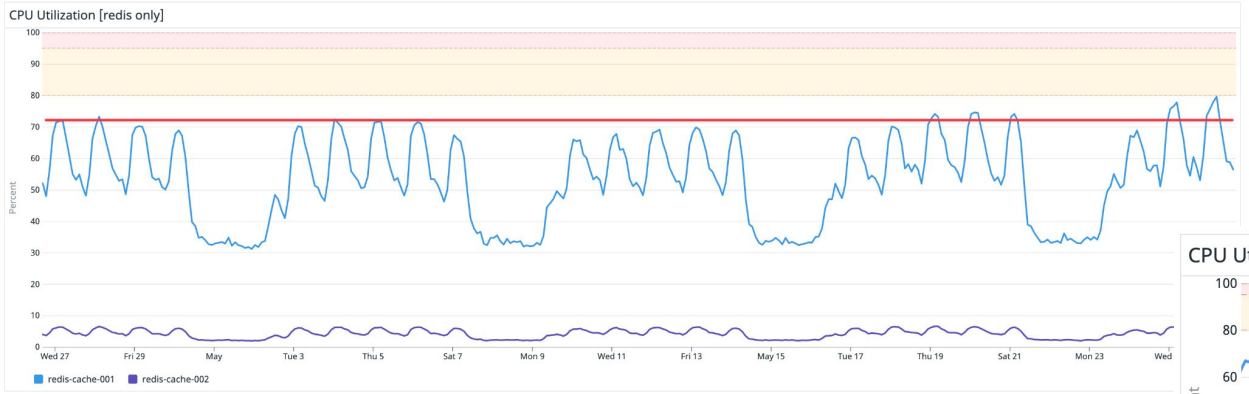
- Caching
- High churn, temporary data

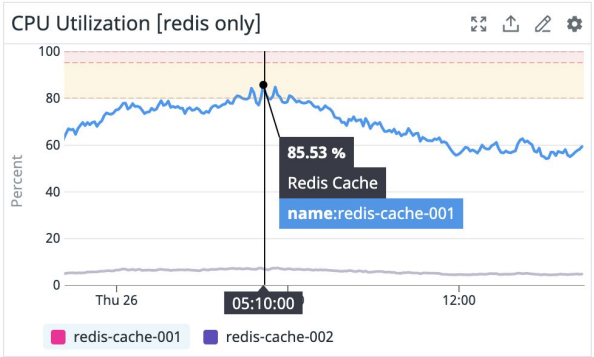
Elevated response times: Redis capping CPU





2 months later...





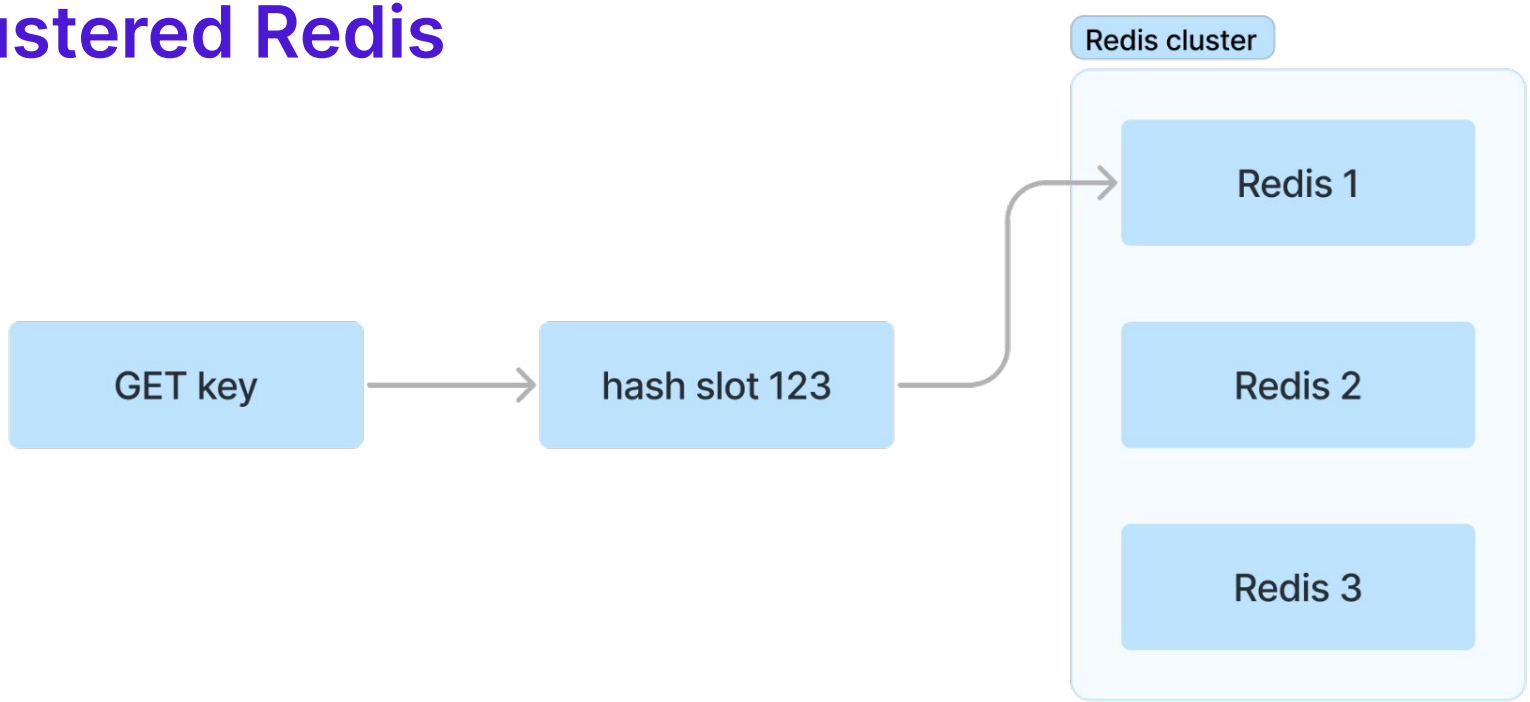
If we can’t make it bigger...

We need more than one!

What are our options?

- Functional sharding?
- Horizontal sharding?

Clustered Redis



*The fine print

Multi-key operations only work if the keys are in the same hash slot:

```
GET key1 key2 key3 # BAD
```

You can force two keys to have the same hash slot by using hashtags:

```
GET {org1}/key1 {org1}/key2 {org1}/key3 # GOOD
```

Is this good enough?

Short version: It’ll work

End up hashing by organization:

```
{:org_id/:purpose}/:id
```

Just do it

Moving the data

Pure cache:

1. Just update the connection string

Source of truth:

1. Write new and old
2. Backfill
3. Read new
4. Stop writing old
5. Drop old

Audit time: What type of data?

- Source of truth
- Source of truth with expiry - skip backfill
- Performance cache - only to speed up requests
- Capacity cache - provides capacity

How do you know if capacity cache?

Short answer: you don’t.

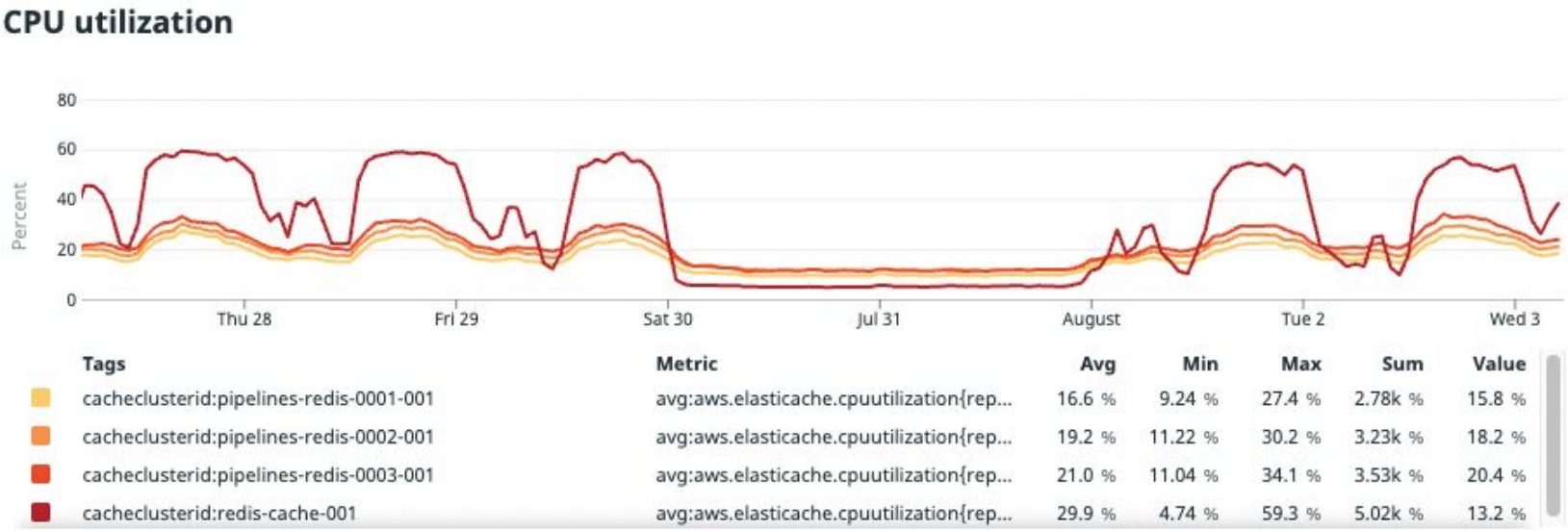
You can make an educated guess:

- Previous incidents
- Chaos engineering
- Load testing

Migrating capacity cache

- Warm the cache
- Percentage based rollout
- Cheaper flags

It’s done! (well, enough)



Conclusions?

Some thoughts

- Investigate what you have
- Remember the goal
- Tell the story

Buildkite