

# Midas

## On-the-fly-Schema-Migration Tool

### For



mongoDB

# Schema Migration Problems

- Applications have to hand-roll their own schema migration infrastructure or use some third-party tool
- Difficult to migrate TBs of data without downtime
  - unacceptable from SLA stand-point!
- How about: on-the-fly schema migration - a *Midas Touch*?

# Zero-downtime Deployment

- Expansion Scripts
  - Apply changes to the documents safely that do not break backwards compatibility with existing version of the application.
  - e.g Adding, copying, merging, splitting fields in a document.
- Contraction Scripts
  - Clean up any database schema that is not needed after the upgrade.
  - e.g. removing fields from a document.

# The Mechanics

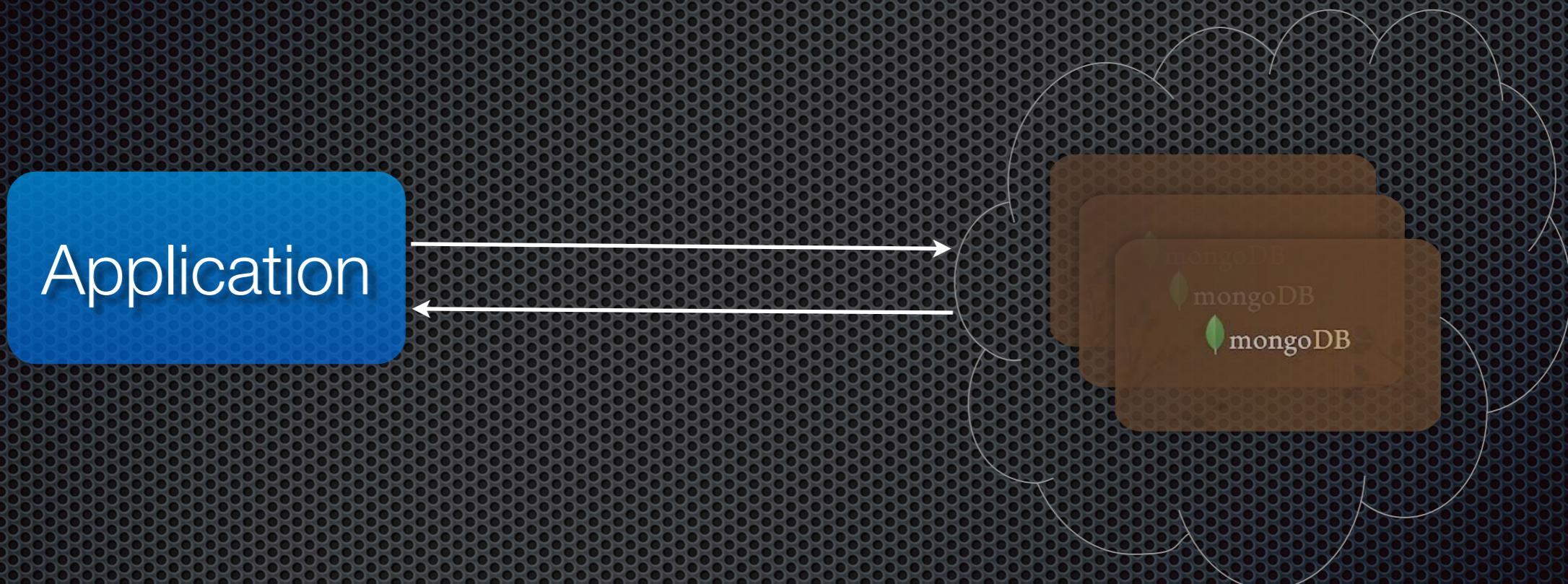
- Run Expansion scripts before upgrading the application
- Upgrade the cluster, a node at a time
- Run Contraction Scripts
  - Once the system has been completely upgraded and deemed stable.
  - Typically, contractions can be run, say days/weeks after complete validation.

# Do we need DB rollback?

- Short Answer
  - No
- Long Answer
  - Reversing DB changes can lead to potential loss of data or leave it in an inconsistent state.
  - Its safer to rollback application without needing to rollback DB changes as expansions are backward compatible.

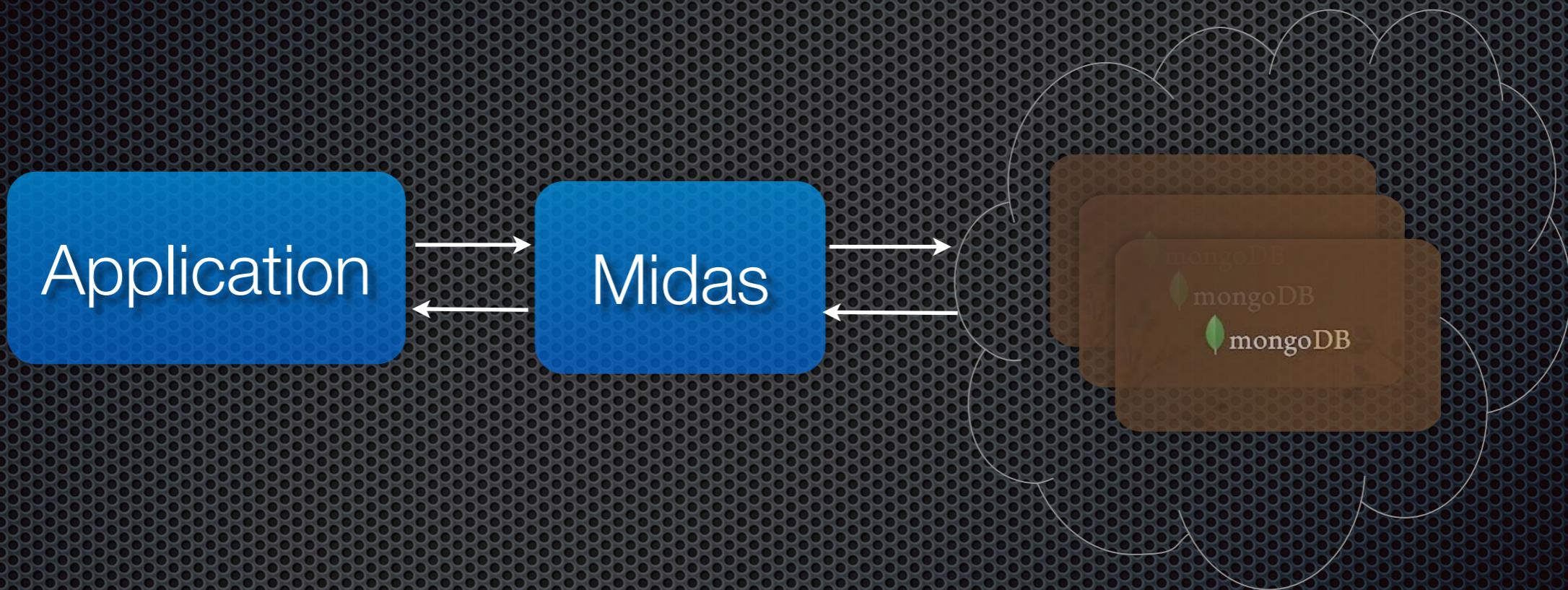
# How it works?

# An Architectural Overview



# How it works?

## An Architectural Overview

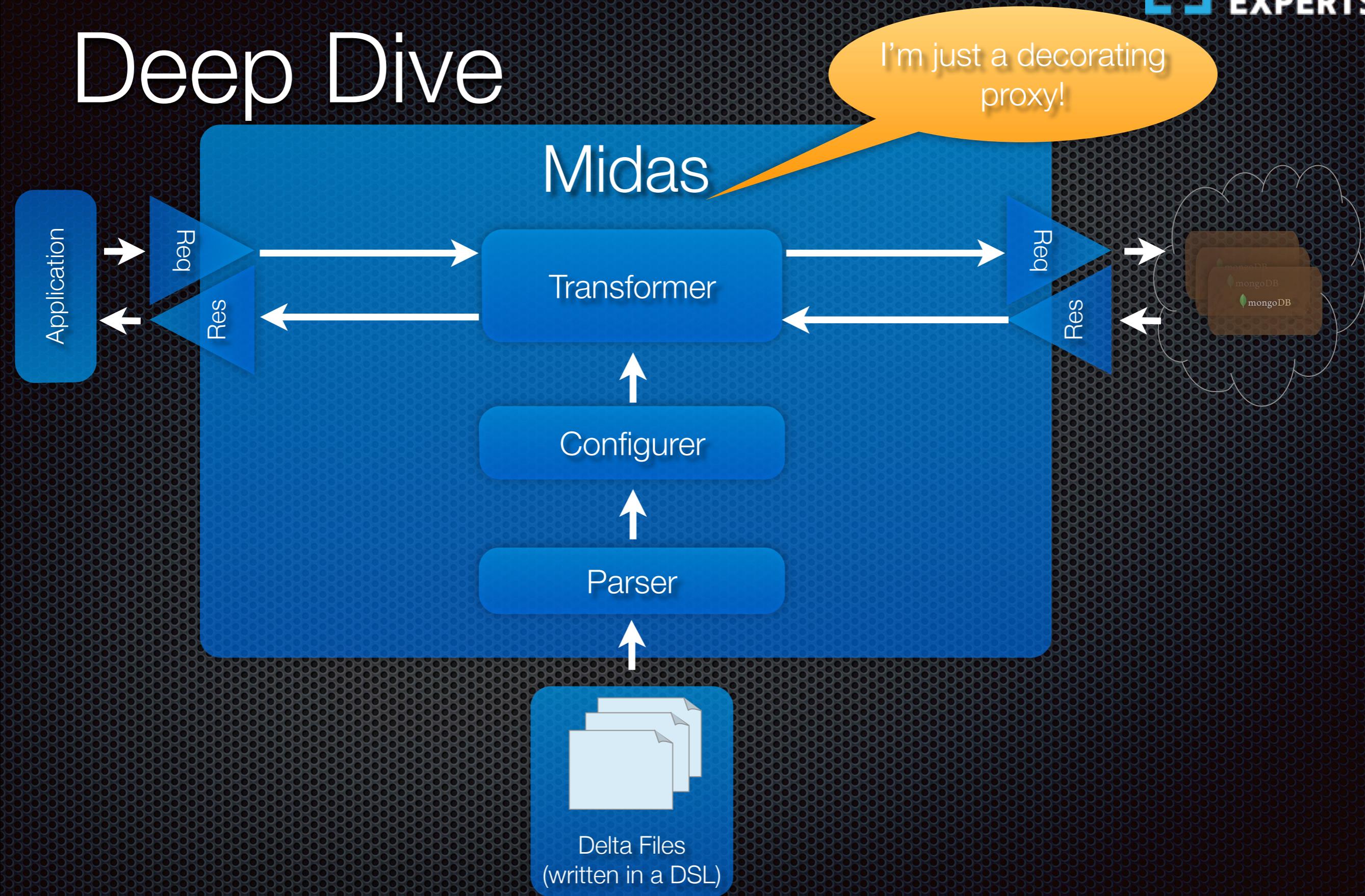


- Intercepts Responses at Protocol Level
- Upgrades/Downgrades Document schema in-transit

# Protocol Level brings Transparency

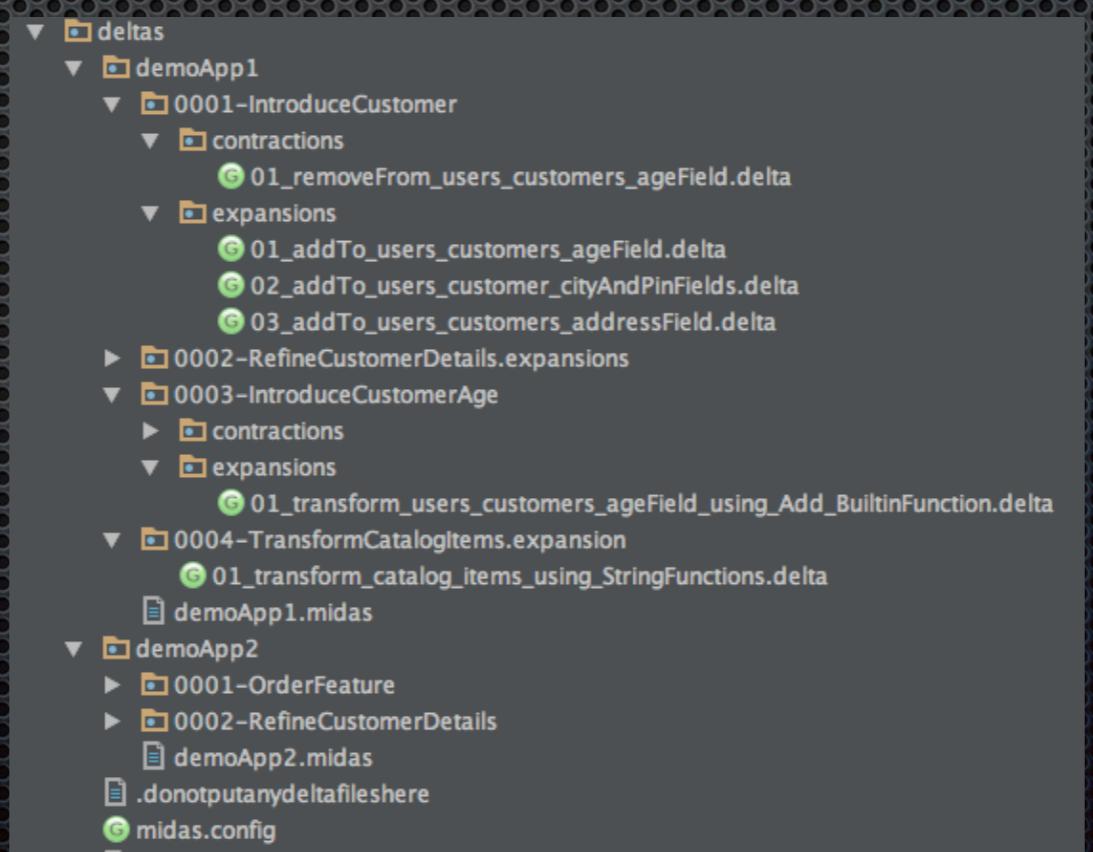
- From the App perspective
  - Midas is Agnostic of Language specific drivers and versions within those languages
  - Works with versions of Ruby, Python, C# and Java drivers.
- From MongoDB perspective
  - Midas is Agnostic of the MongoDB configurations
  - Works with Standalone, Replica Sets, Sharded environments.

# Deep Dive



# Delta scripts

- Convention over Configuration
  - Group ChangeSet as Directories
  - Group Expansion and Contraction as Directories within ChangeSet
  - Scripts are ordered starting with 0001\_xxx.delta as the first delta



# Delta scripts

- Each Delta script is written using a DSL
  - Very close to MongoDB lingo, virtually no learning curve.

```
use users
db.customers.remove('["address.line1"]')
db.customers.merge(["lname", "fname"], "", "name")
```

```
use transactions
db.orders.add('dispatch : { status: \'NOT DELIVERED\' }')
```

# Agile App Delivery & DevOps

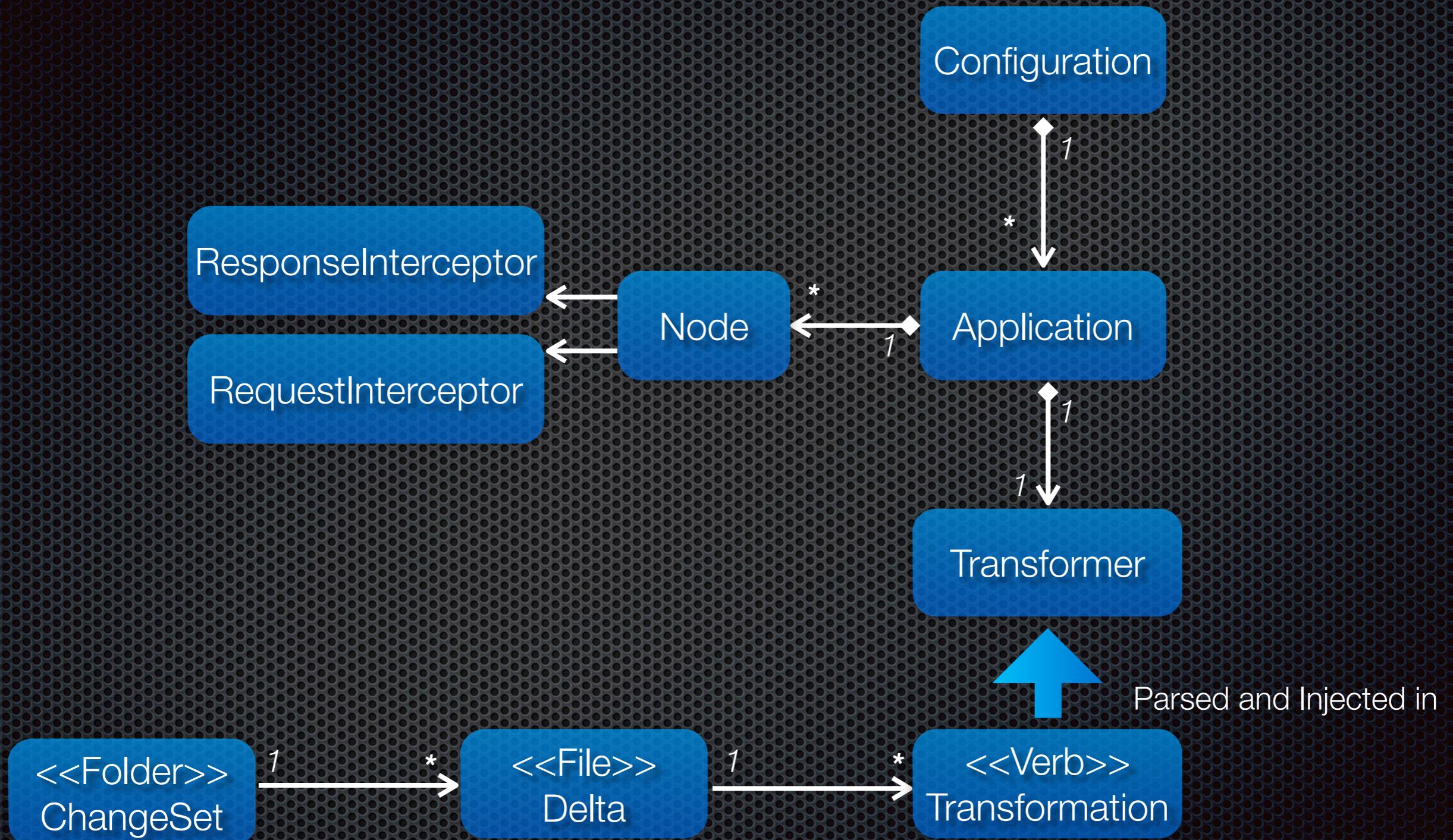
- Inject Midas into Architecture
  - Start or Middle of project
- Supports Development of Application in small-steps
  - Add Application ChangeSets/Deltas on-the-fly
- Copes with Load
  - Add/Remove Application Nodes on-the-fly
- Supports Multiple Applications
  - Add/Remove Application(s) on-the-fly

# No Leaky Abstractions

- Does not expect the Domain Model to be aware of versioning.
  - Allows developers to focus on the domain while freeing them from versioning concerns.
  - If you wish to take charge, Midas will not come in your way.
- Midas maintains versioning information within the document itself.
  - `_expansionVersion` and `_contractionVersion` fields are part of the documents “touched” by Midas.
  - Updates them during request and response.



# Domain Model



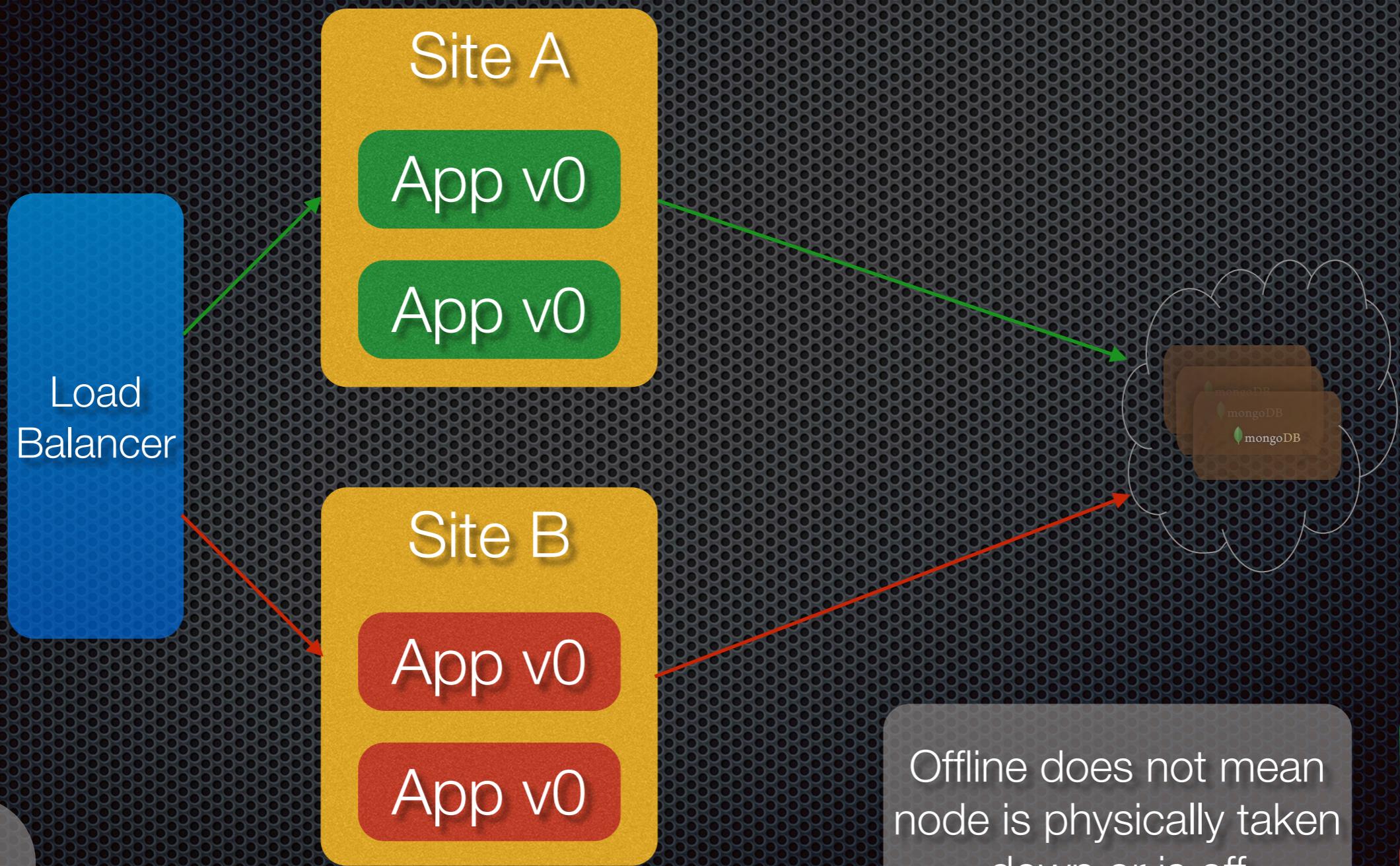
# Caveats

- Never ever change a delta that has been applied to production.
  - Always move forward in time.
  - Reverse a change by a counter-change
- Force-update migration on documents not
  - expanded by App demand, and then proceed to contraction
  - contracted by App demand, and then proceed to next App-upgrade cycle

# Zero-Downtime Deployment Configuration - 1

## Injecting and Using Midas

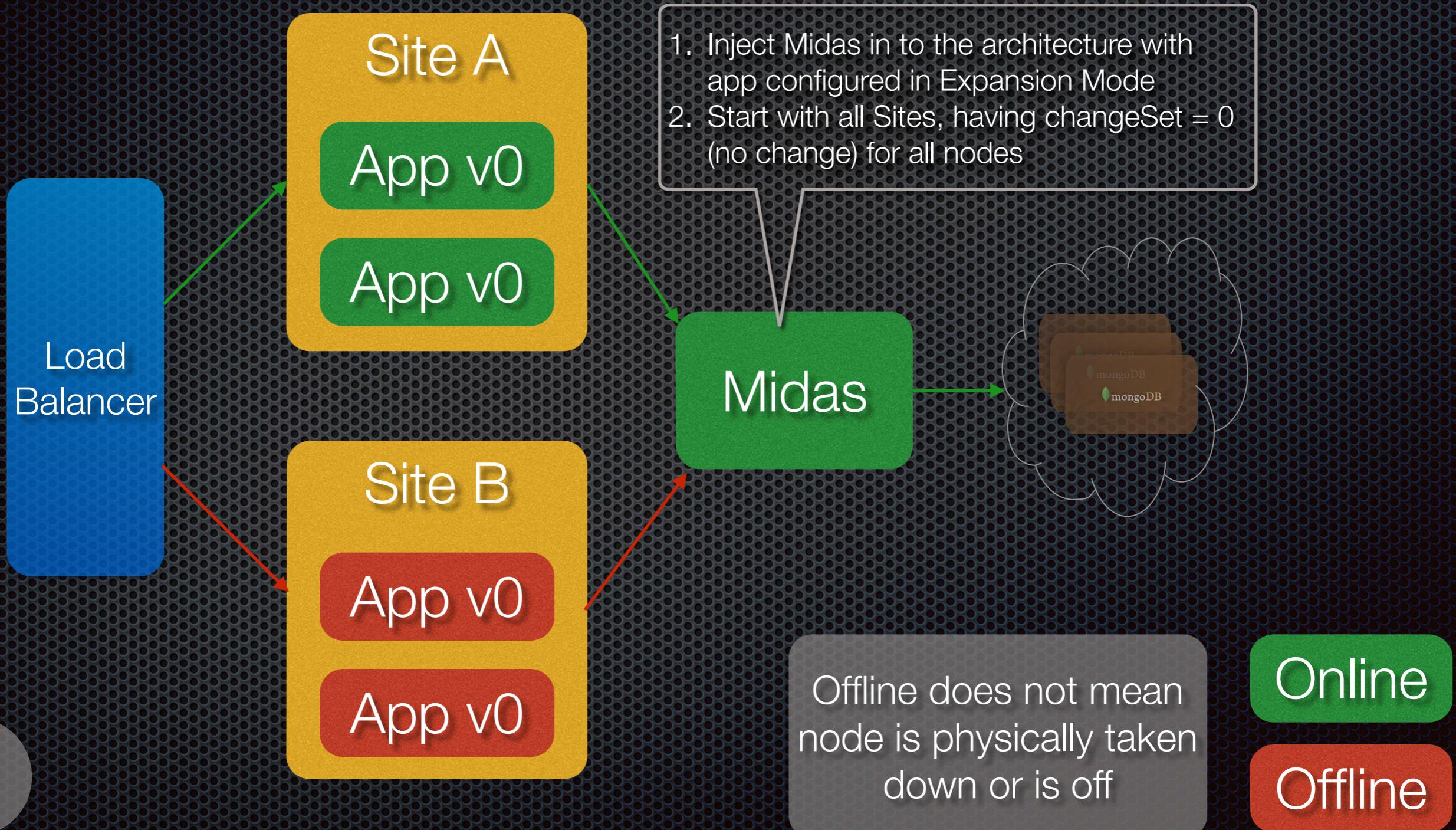
# Deployment config - 1



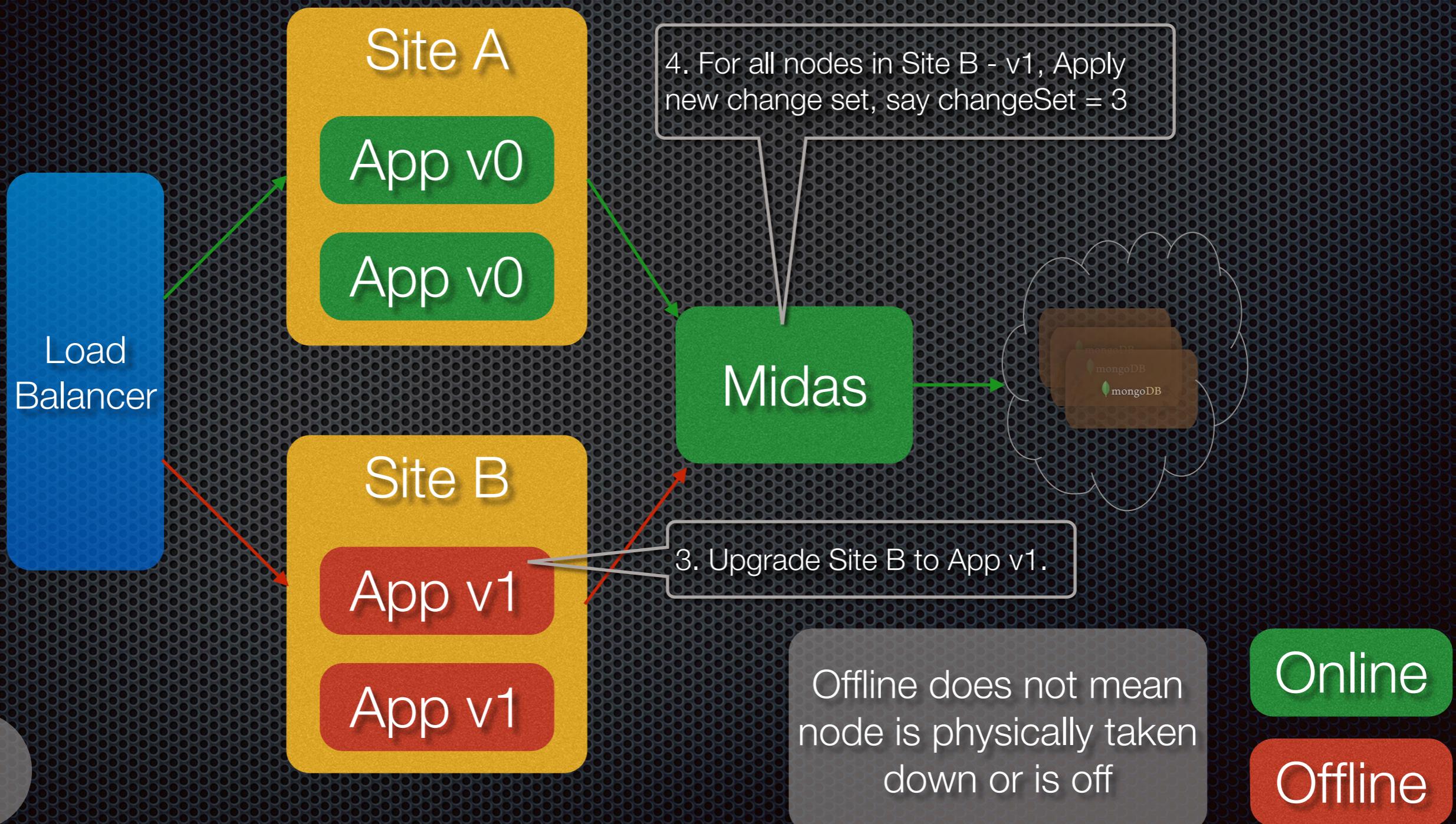
Offline does not mean node is physically taken down or is off

Online  
Offline

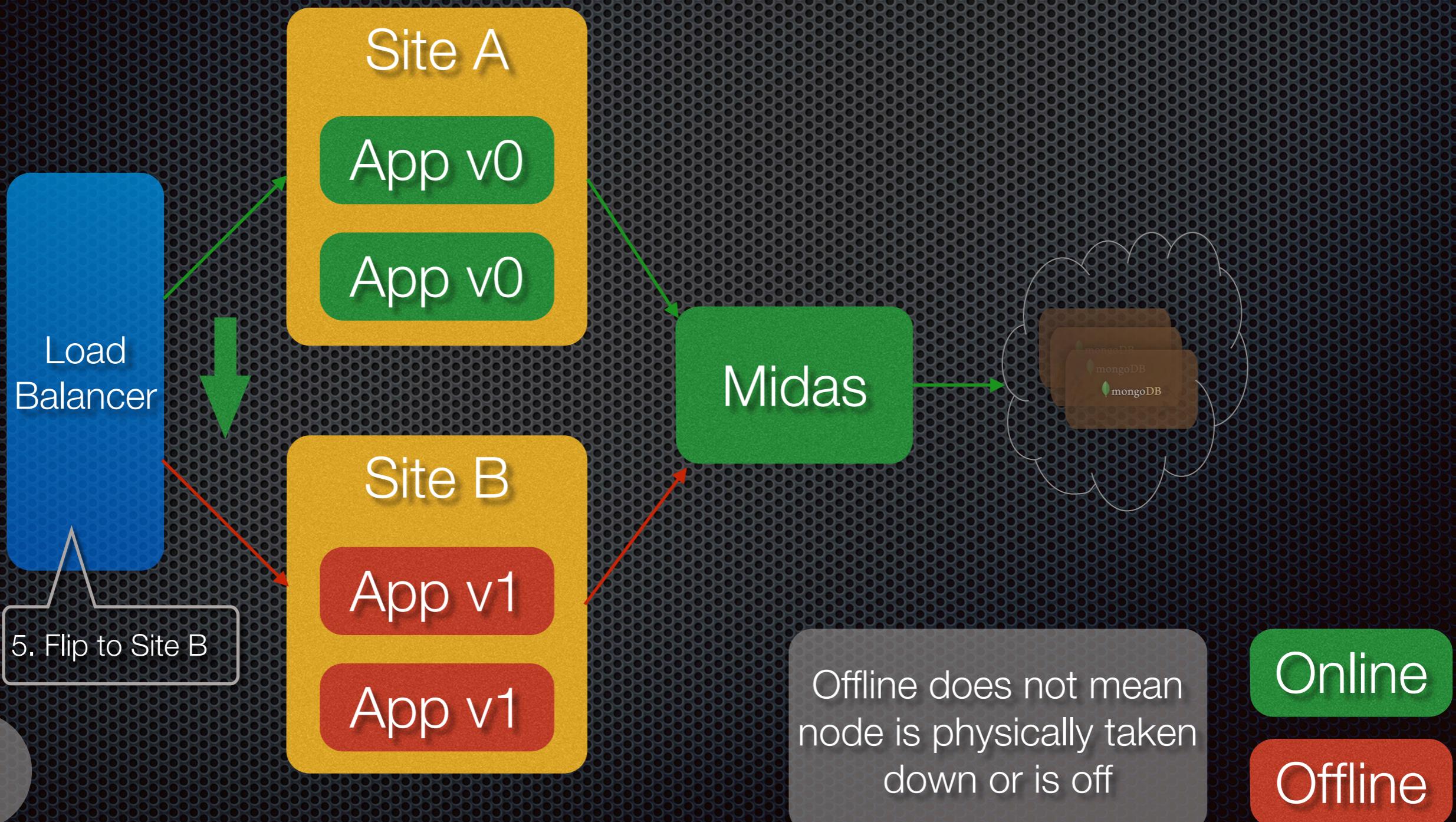
# Deployment config - 1



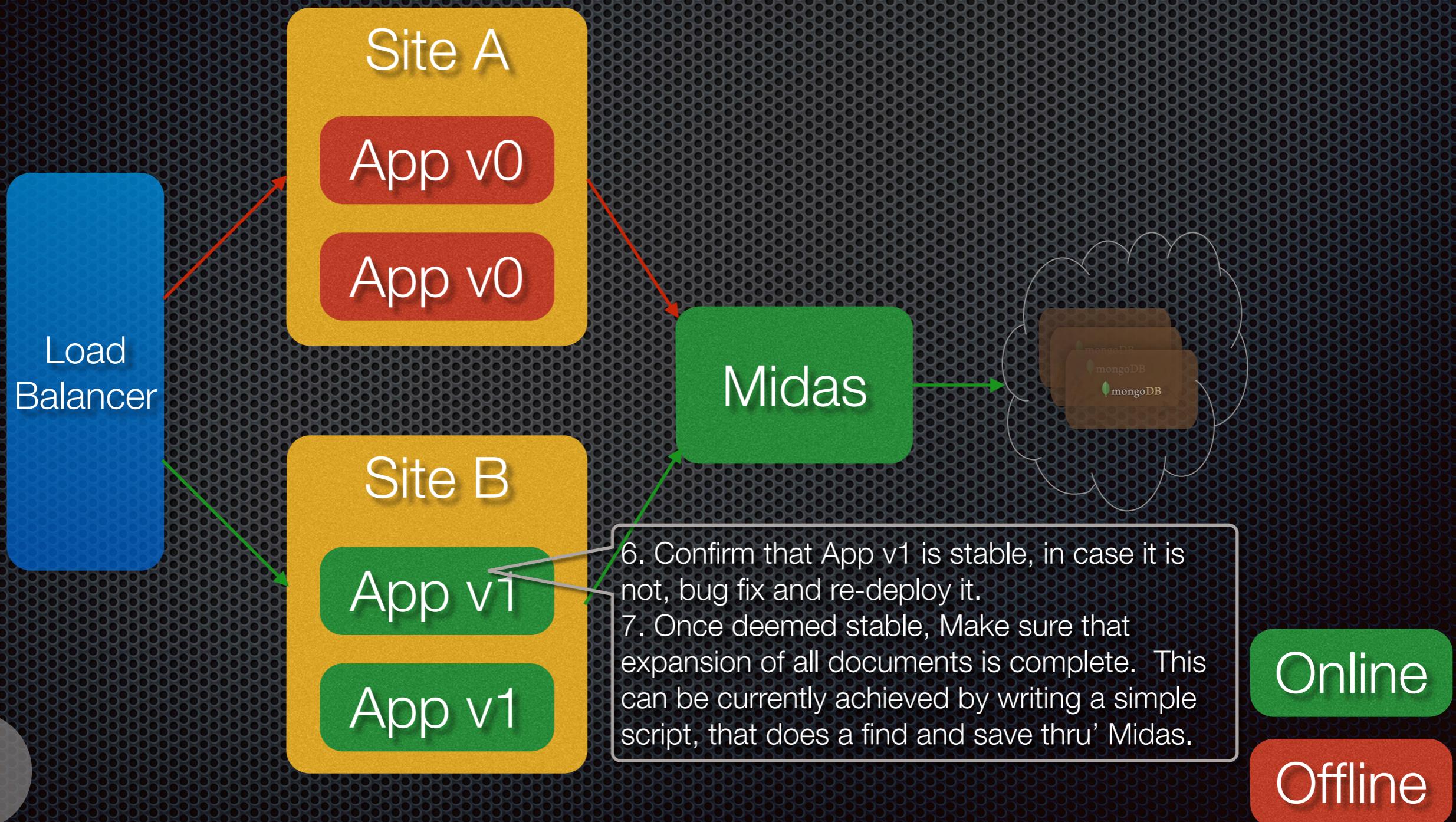
# Deployment config - 1



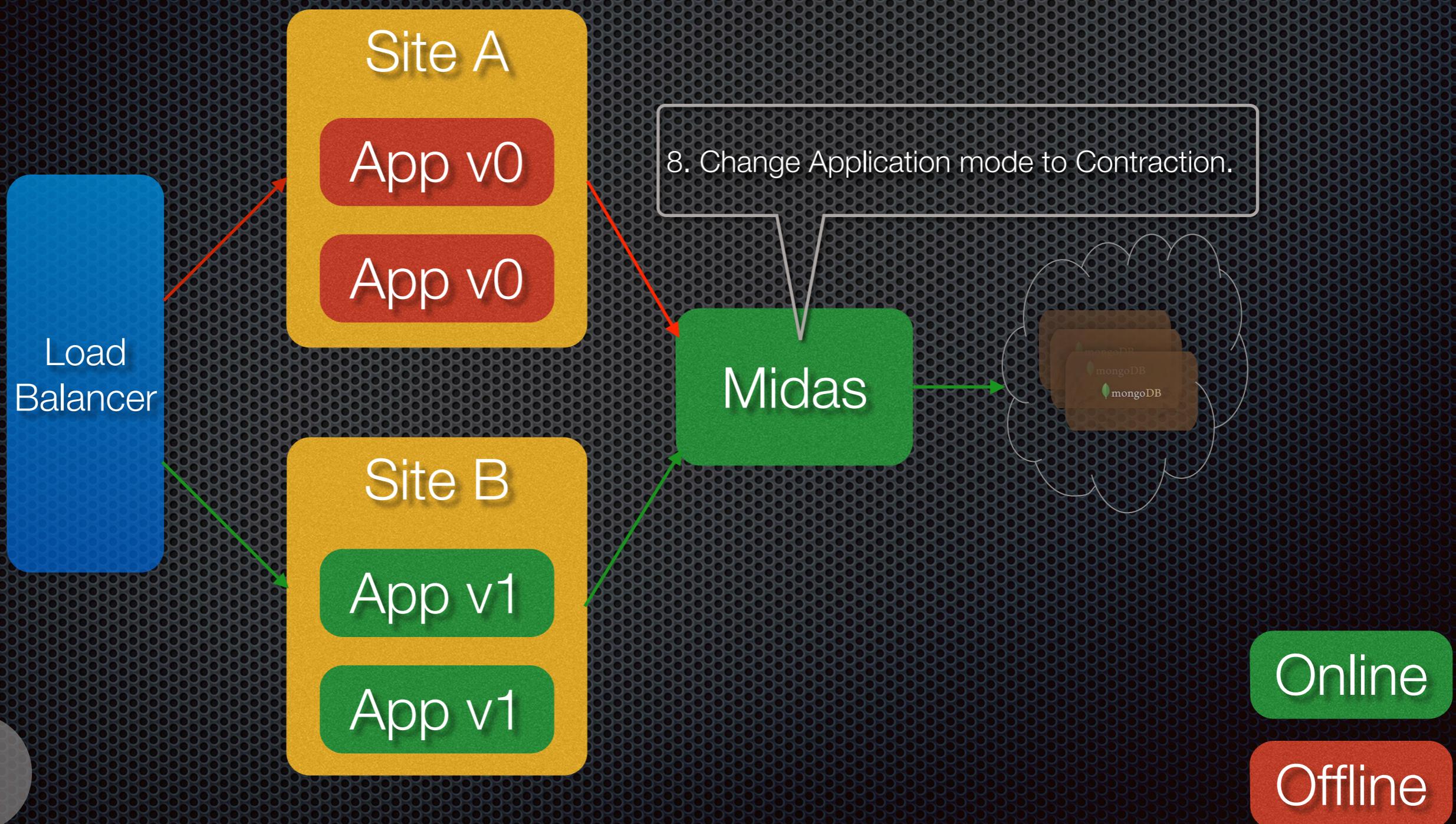
# Deployment config - 1



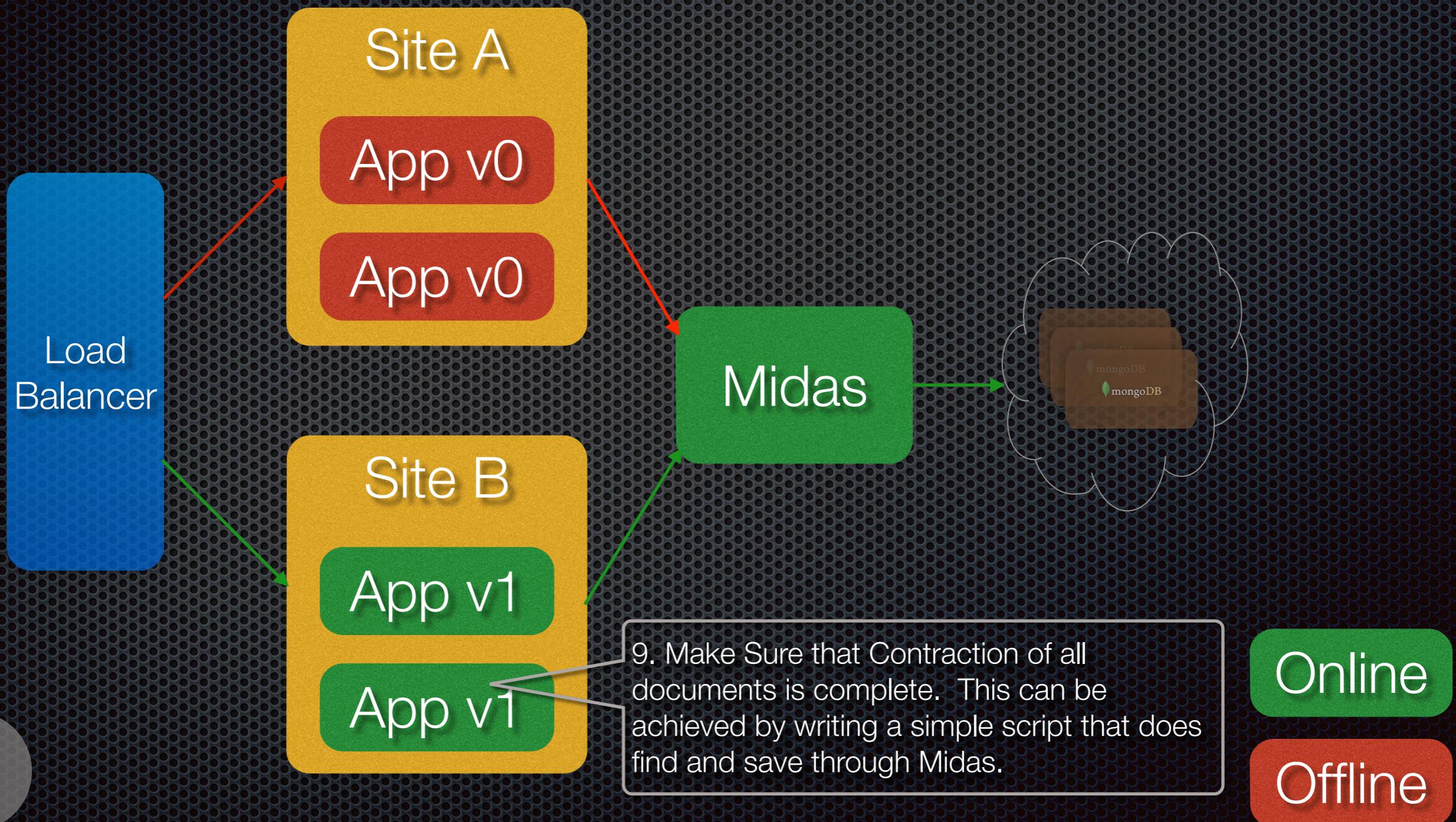
# Deployment config - 1



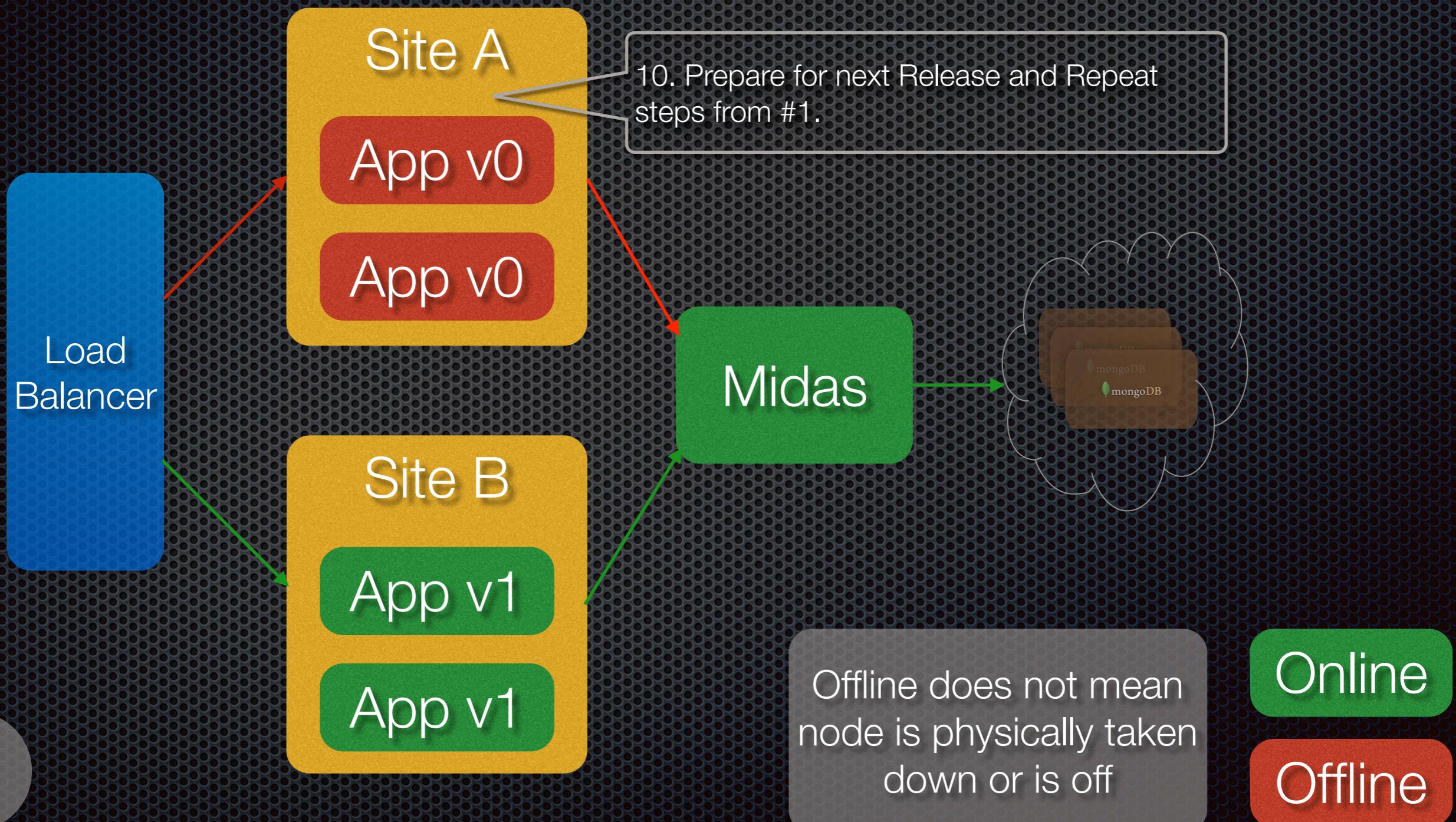
# Deployment config - 1



# Deployment config - 1



# Deployment config - 1

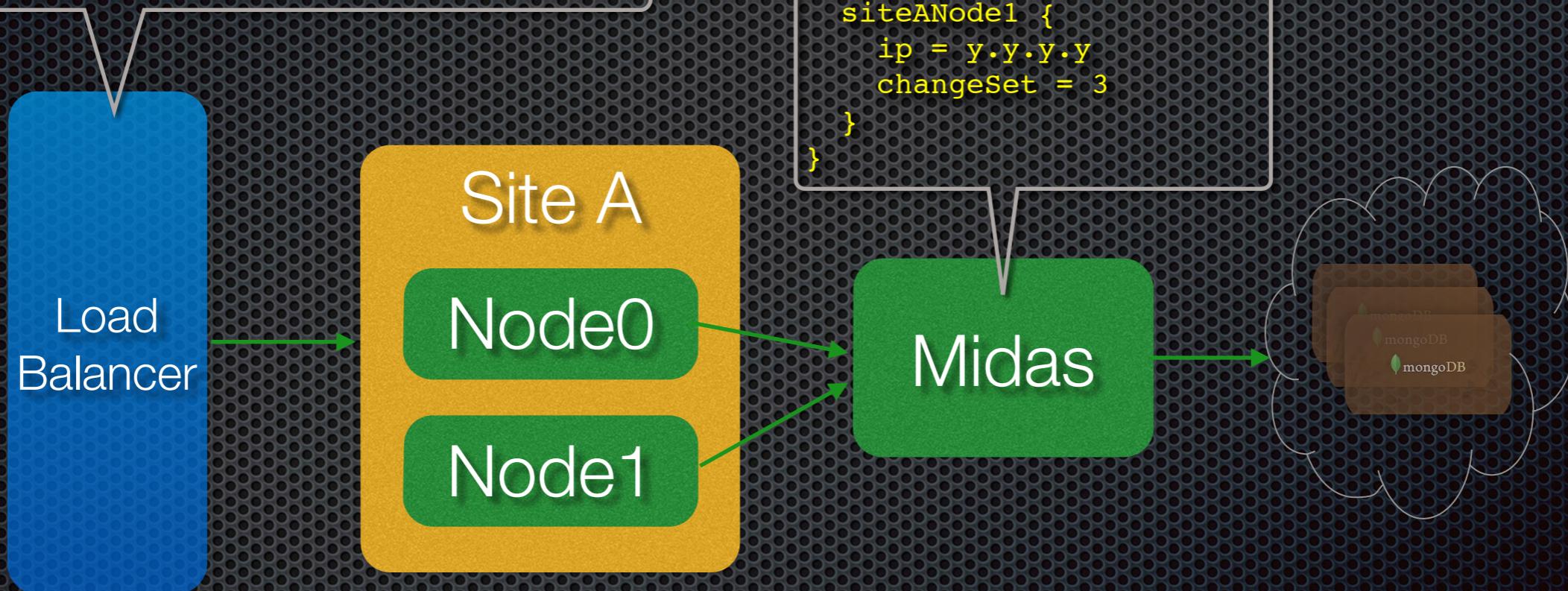


# Zero-Downtime Deployment Configuration - 1

Node Removal at Runtime

Say, due to some problems, you want to remove Node0 from service:

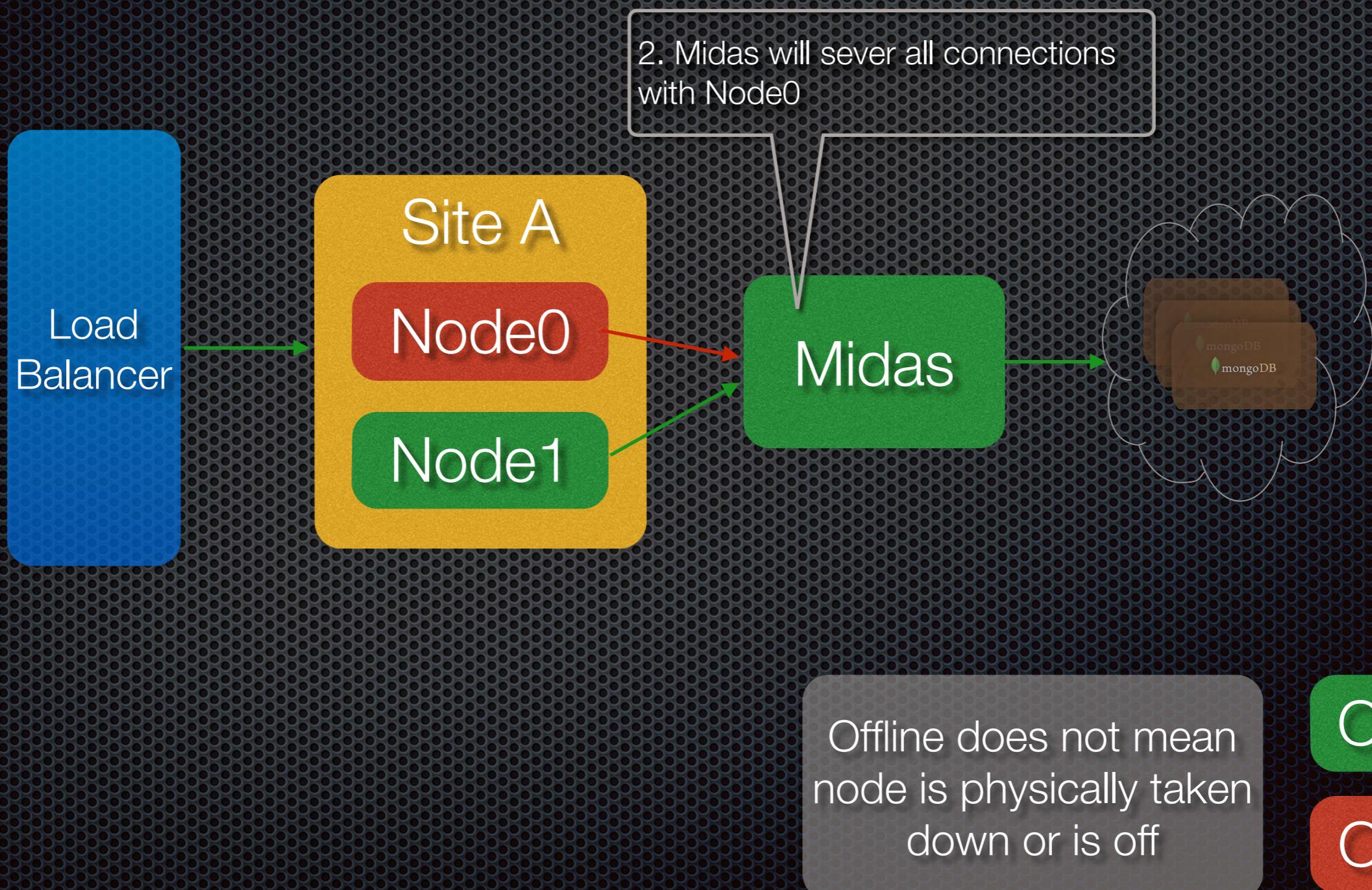
1. Remove Node0 from LB.



2. Remove Node0 from **app.midas**

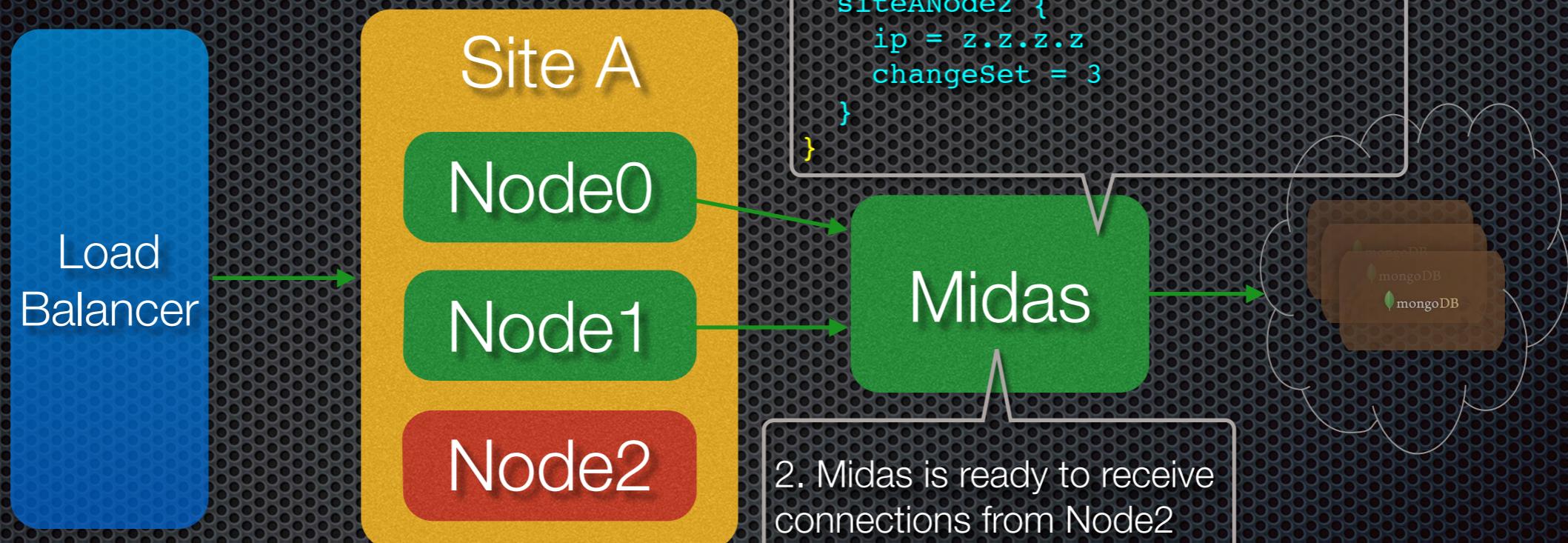
```

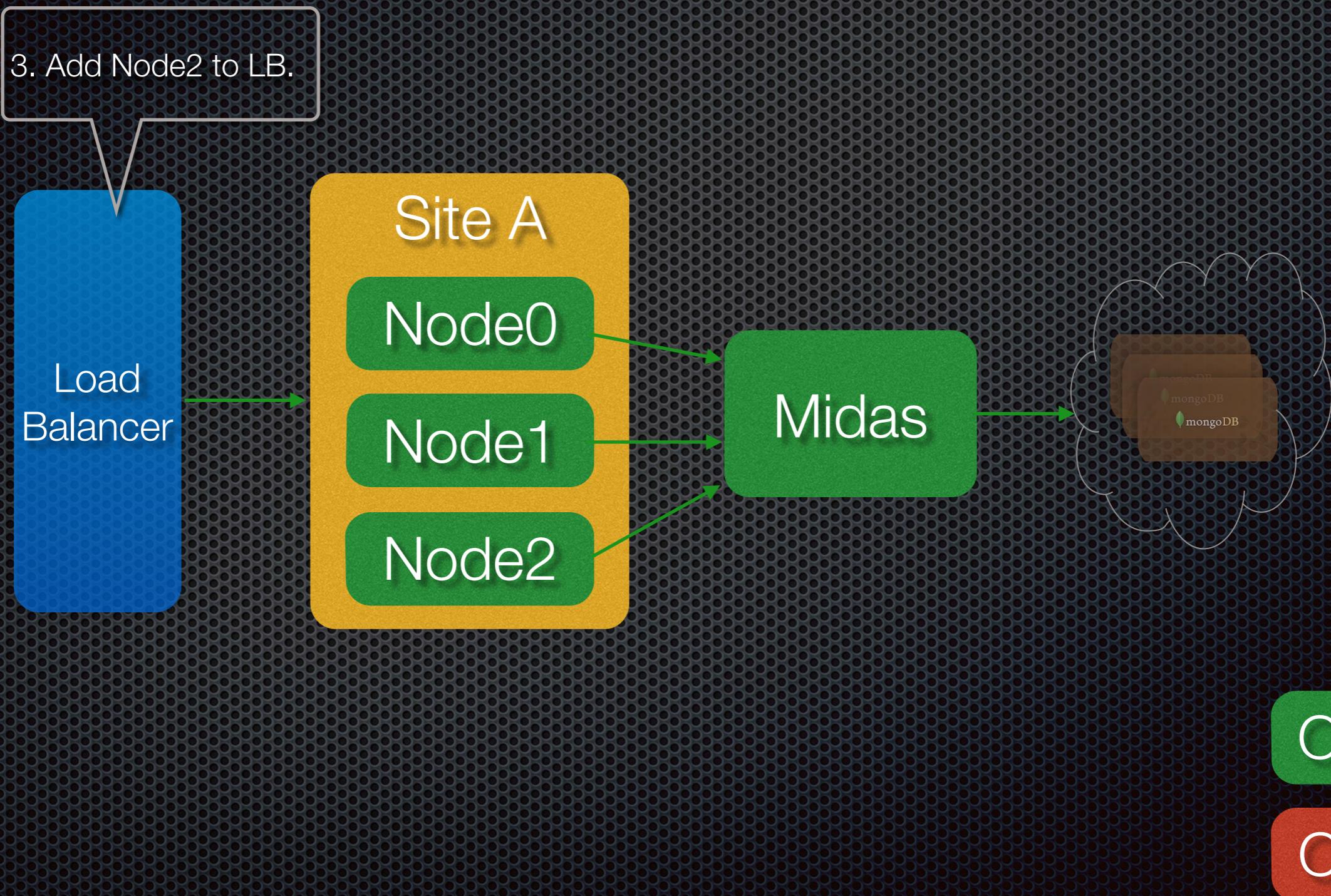
demoAppV1 {
  mode = contraction
  // siteANode0 {
  //   ip = x.x.x.x
  //   changeSet = 3
  // }
  siteANode1 {
    ip = y.y.y.y
    changeSet = 3
  }
}
  
```



# Zero-Downtime Deployment Configuration - 1

Node Injection at Runtime

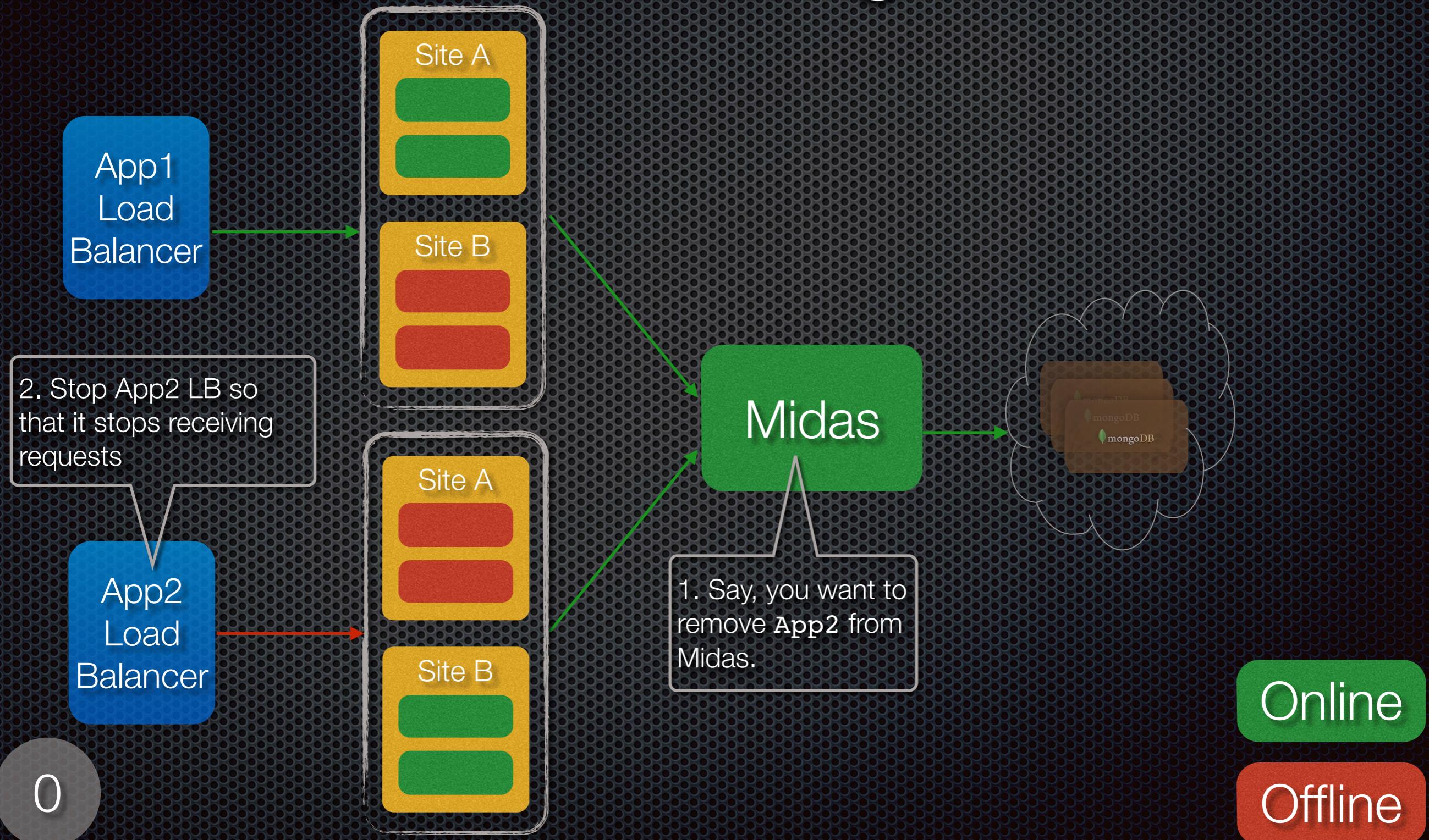


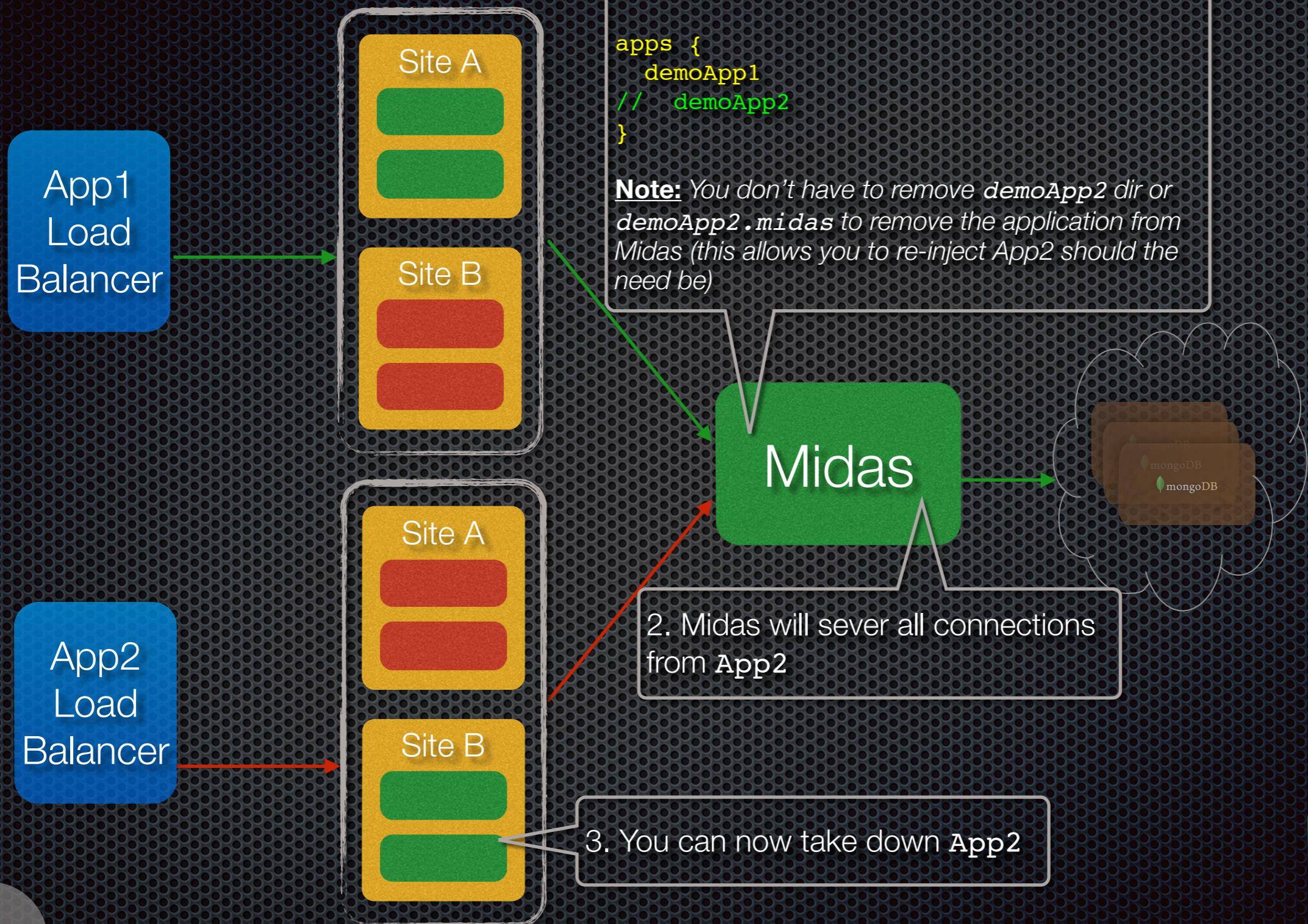


# Zero-Downtime Deployment Configuration - 1

App Removal at Runtime

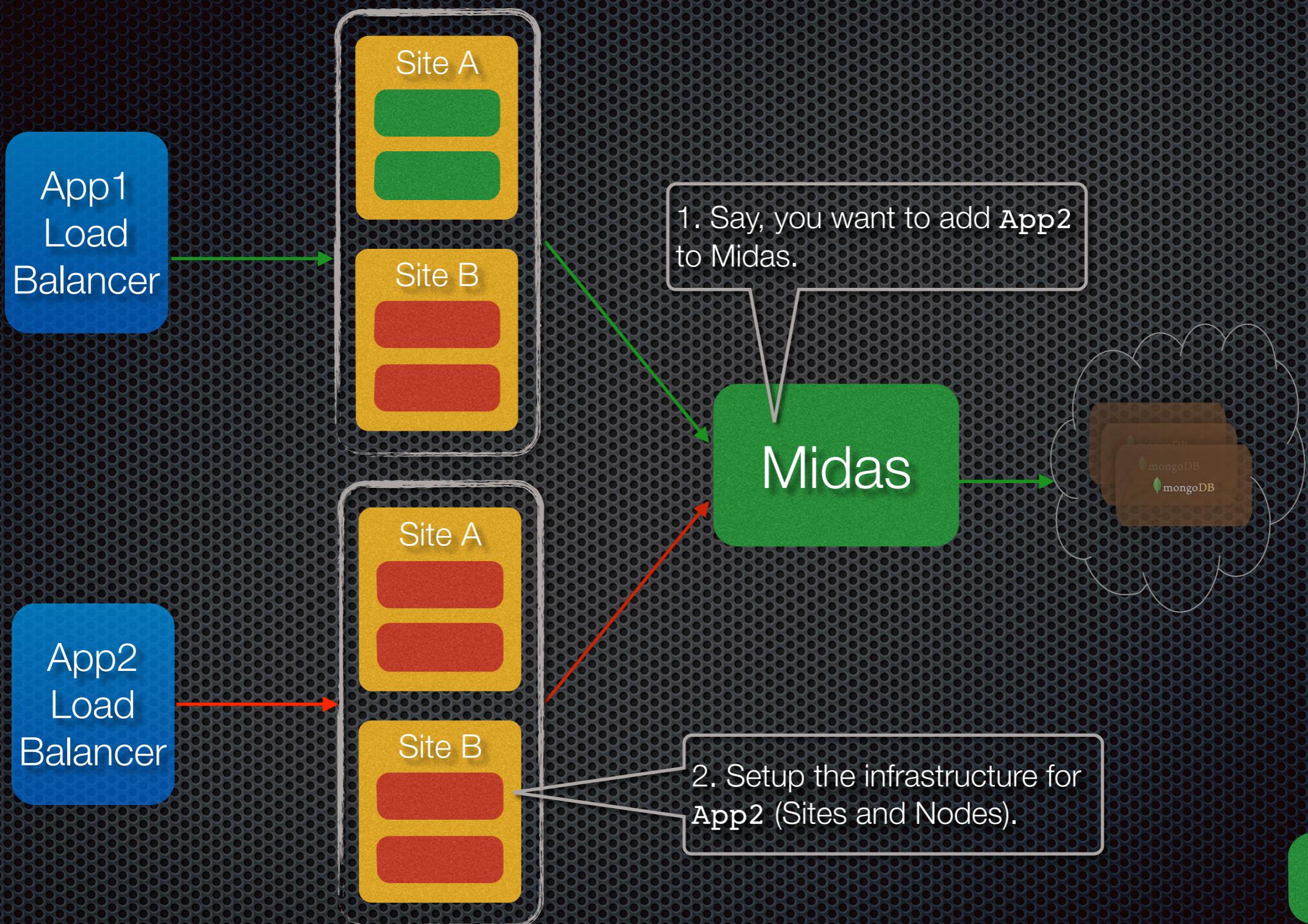
# Deployment config - 1

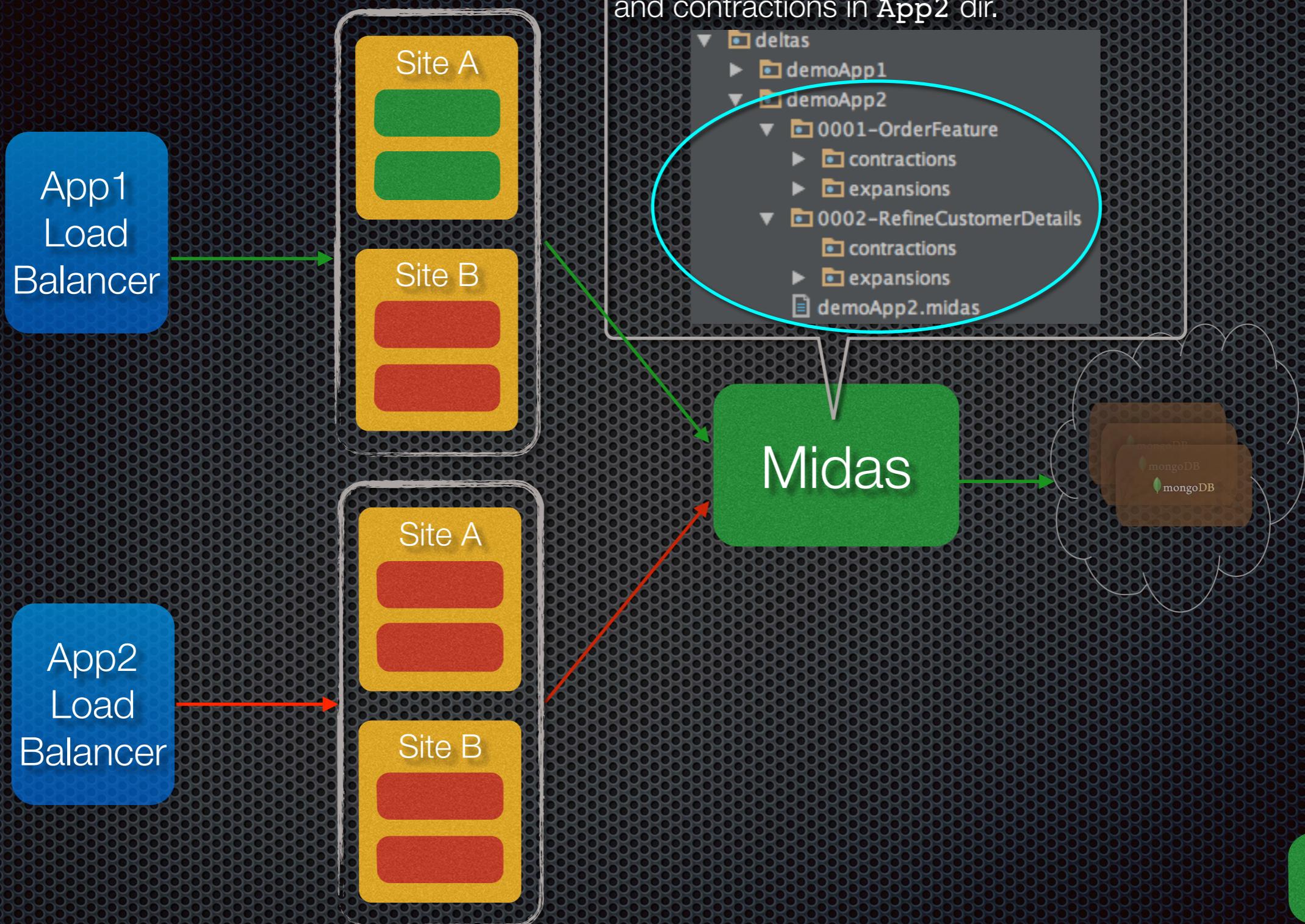


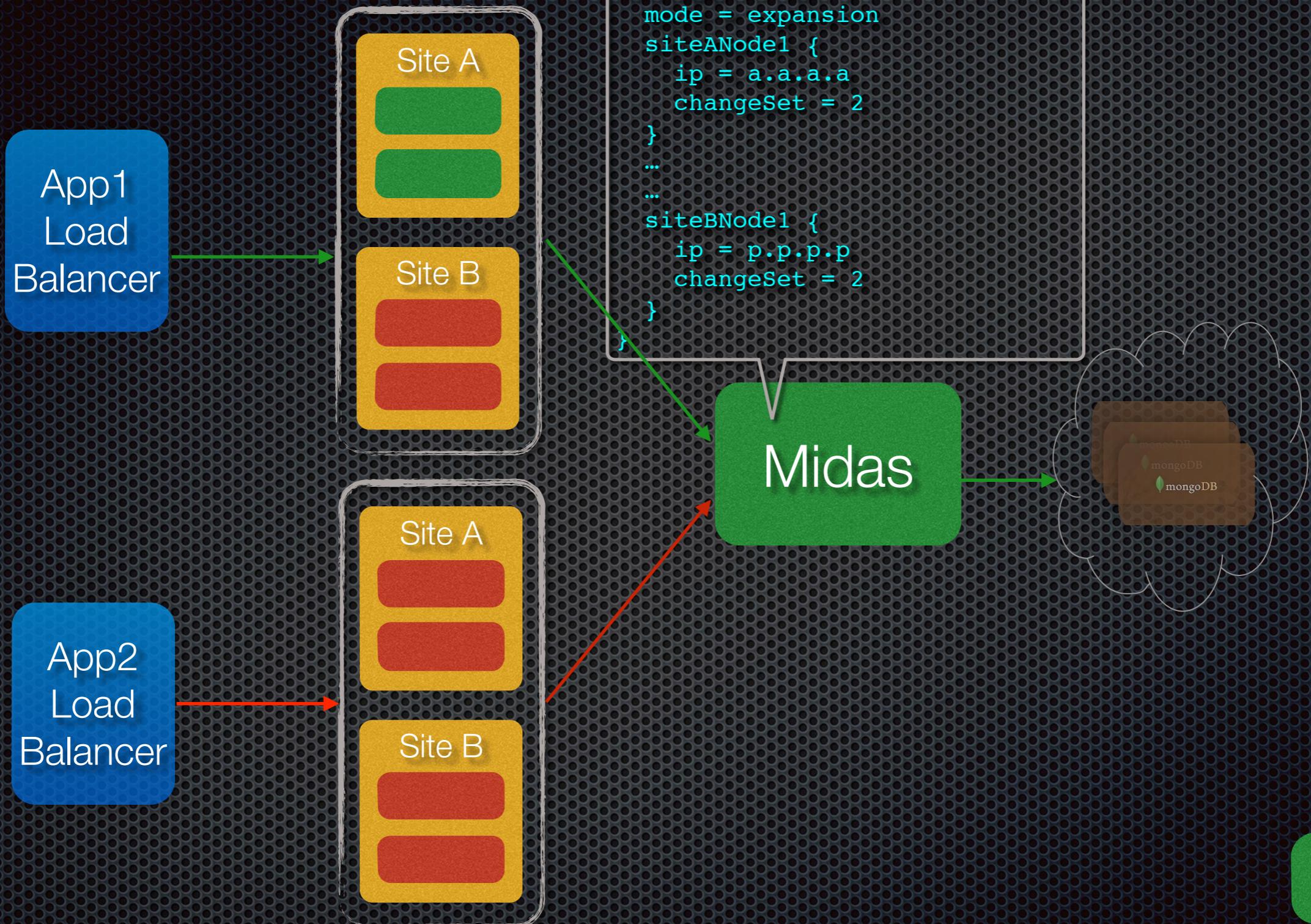


# Zero-Downtime Deployment Configuration - 1

## App Injection at Runtime







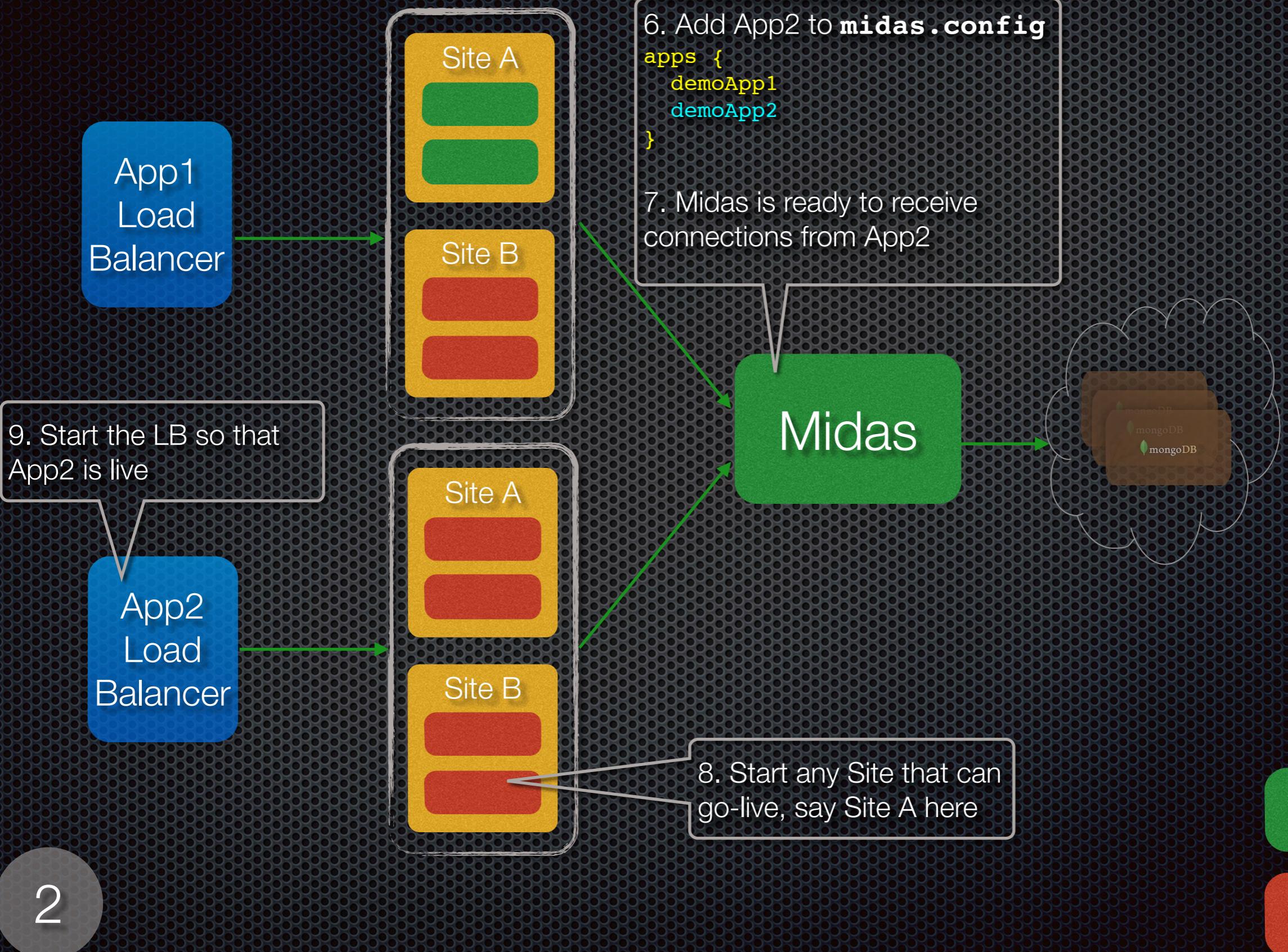
5. Add **demoApp2.midas** (with mode and nodes configured)

```
demoApp2Ver1_0 {  
    mode = expansion  
    siteANode1 {  
        ip = a.a.a.a  
        changeSet = 2  
    }  
    ...  
    ...  
    siteBNode1 {  
        ip = p.p.p.p  
        changeSet = 2  
    }  
}
```

Midas

Online

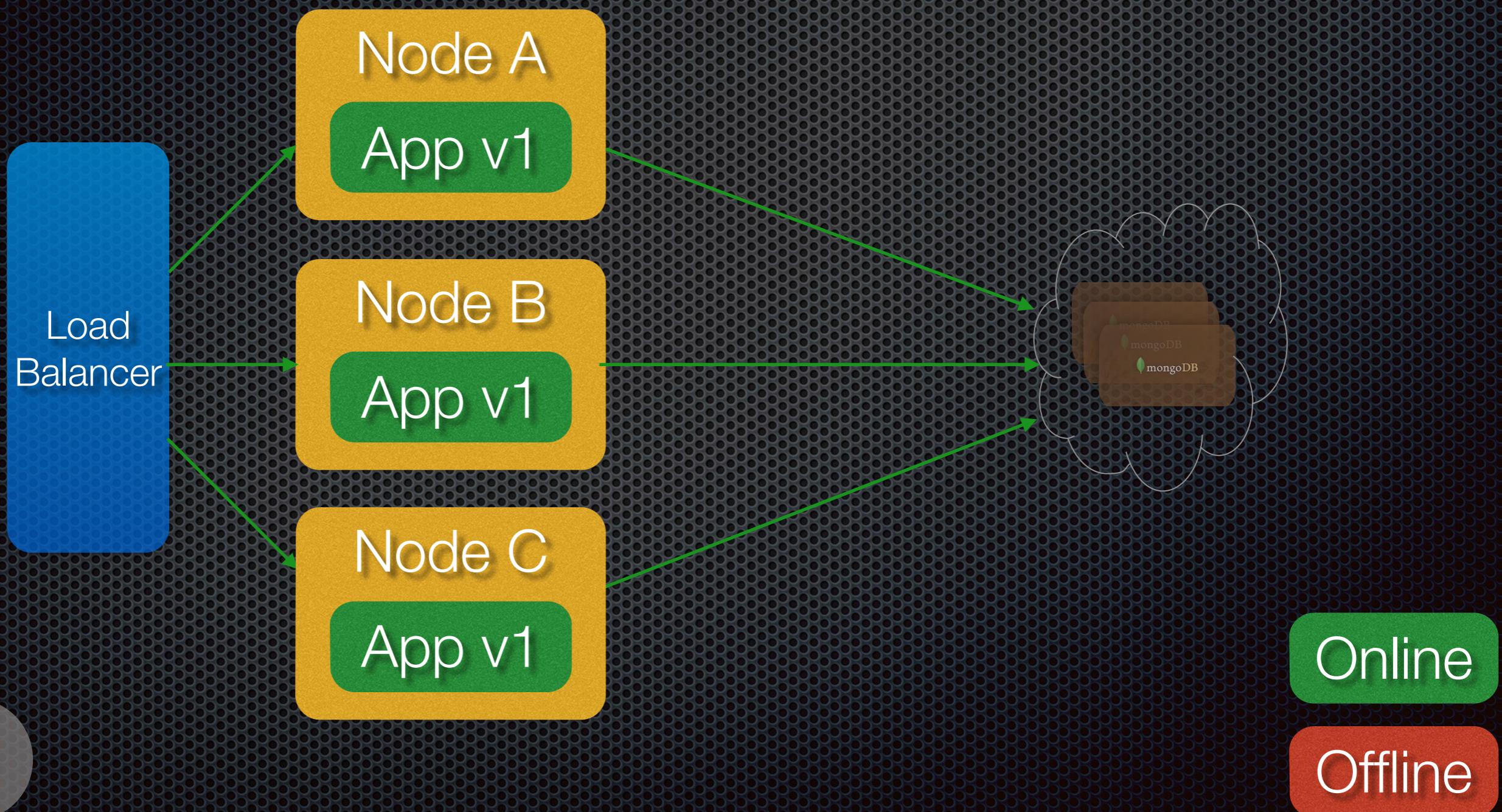
Offline



# Zero-Downtime Deployment Configuration - 2

**Not Supported in this Release**

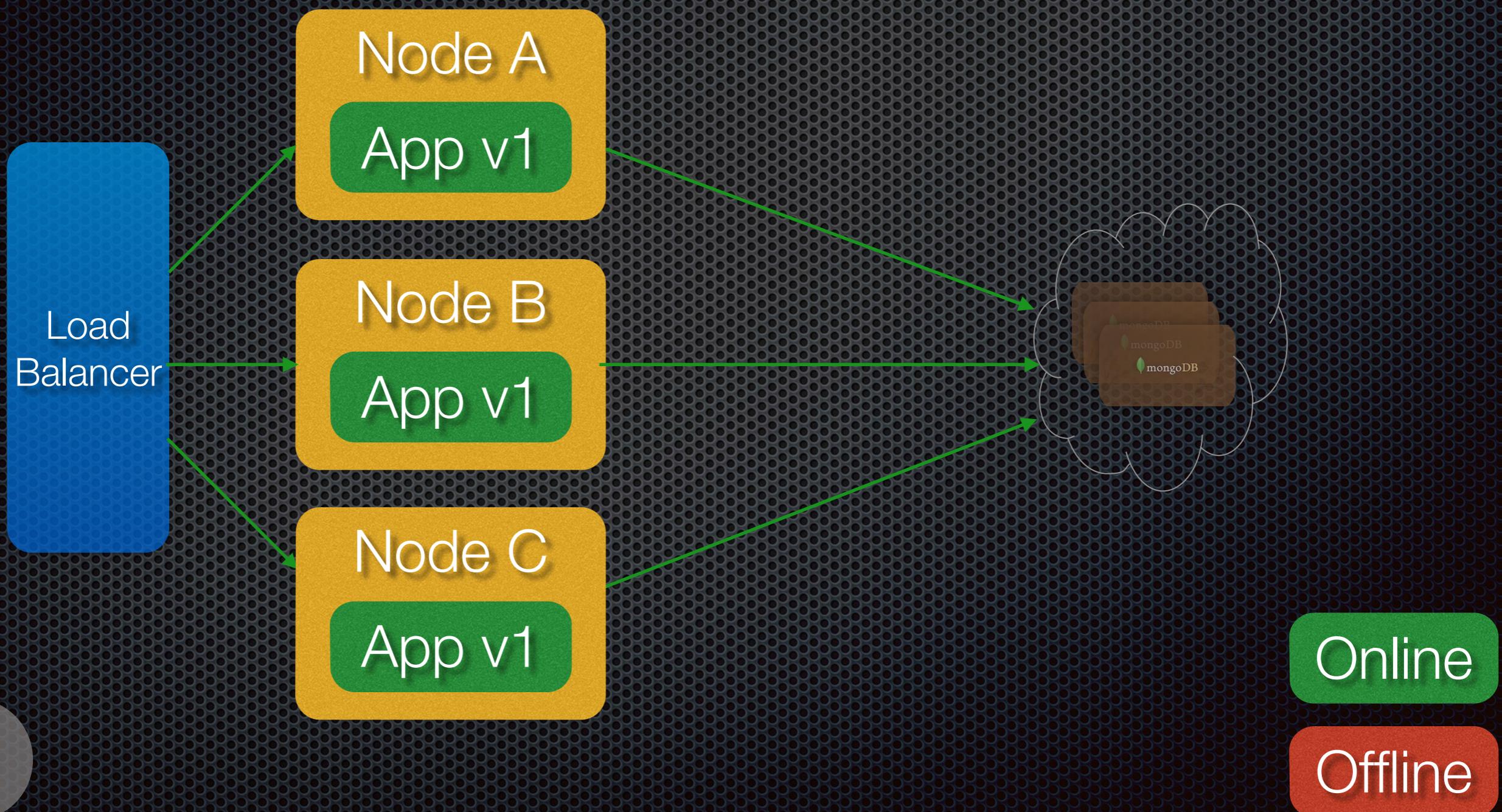
# Deployment config - 2



# Zero-Downtime Deployment Configuration - 3 AWS

**Not Supported in this Release**

# Deployment config - 3



# TODOs

- Midas force migrates documents for Contractions without affecting performance [P1] - Park
- Introduce a Midas by pass mode. [P1]
  - Scenario: After an upgrade cycle (expansion + contraction), performance should not suffer. - Not Required
- Soak Testing [P1] - In Progress
- Load Test [P1] - Will Start This Week
- Performance Metrics [P1] - Will do again 3rd time
- An Admin Client [P2] - Park
  - Force update for documents not expanded/contracted

# The Team

- Brian Blignaut
  - [bblignaut@equalexperts.com](mailto:bblignaut@equalexperts.com)
- Dhaval Dalal [@softwareartisan]
  - [ddalal@equalexperts.com](mailto:ddalal@equalexperts.com)
- Vivek Dhapola
  - [vdhapola@equalexperts.com](mailto:vdhapola@equalexperts.com)
- Komal Jain
  - [kjain@equalexperts.com](mailto:kjain@equalexperts.com)

# References

- Owen Rogers
- <http://exortech.com/blog/2009/02/01/weekly-release-blog-11-zero-downtime-database-deployment/>

# Thank-You