Zero downtime database deployments

Moving to a safer release cadence, based on smaller, more frequent deployments, normally requires deployments to occur while the systems are running. That's scary. In this session we'll discuss how to do it safely.

Alex Yates

Database DevOps Consultant

alex.yates@dlmconsultants.com
@_AlexYates_
workingwithdevs.com

MVP Microsoft® Most Valuable Professional

DATA RELAY

FRIEND OF redgate 2019

@_AlexYates_
#DevOps

Zero Downtime Database Deployments

DLM Consultants

Zero downtime release patterns

Assumptions…

- Source control
- Automated deploy pipelines
- Infra as code

deploy to prod...

@_AlexYates_

#DevOps

Zero Downtime Database Deployments

DLM Consultants

over loaded deploys

@_AlexYates_

#DevOps

Zero Downtime Database Deployments

DLM Consultants

"If we want more changes, we need more deployments."

Chuck Rossi, Facebook
DevOps Handbook, p154

"Deployment need not expose customers to a new version of your service."

Deploy != Release

Art Gillespie, Deploy != Release

blog.turbinelabs.io/deploy-not-equal-release-part-one-4724bc1e726b

Release patterns

Environment-based

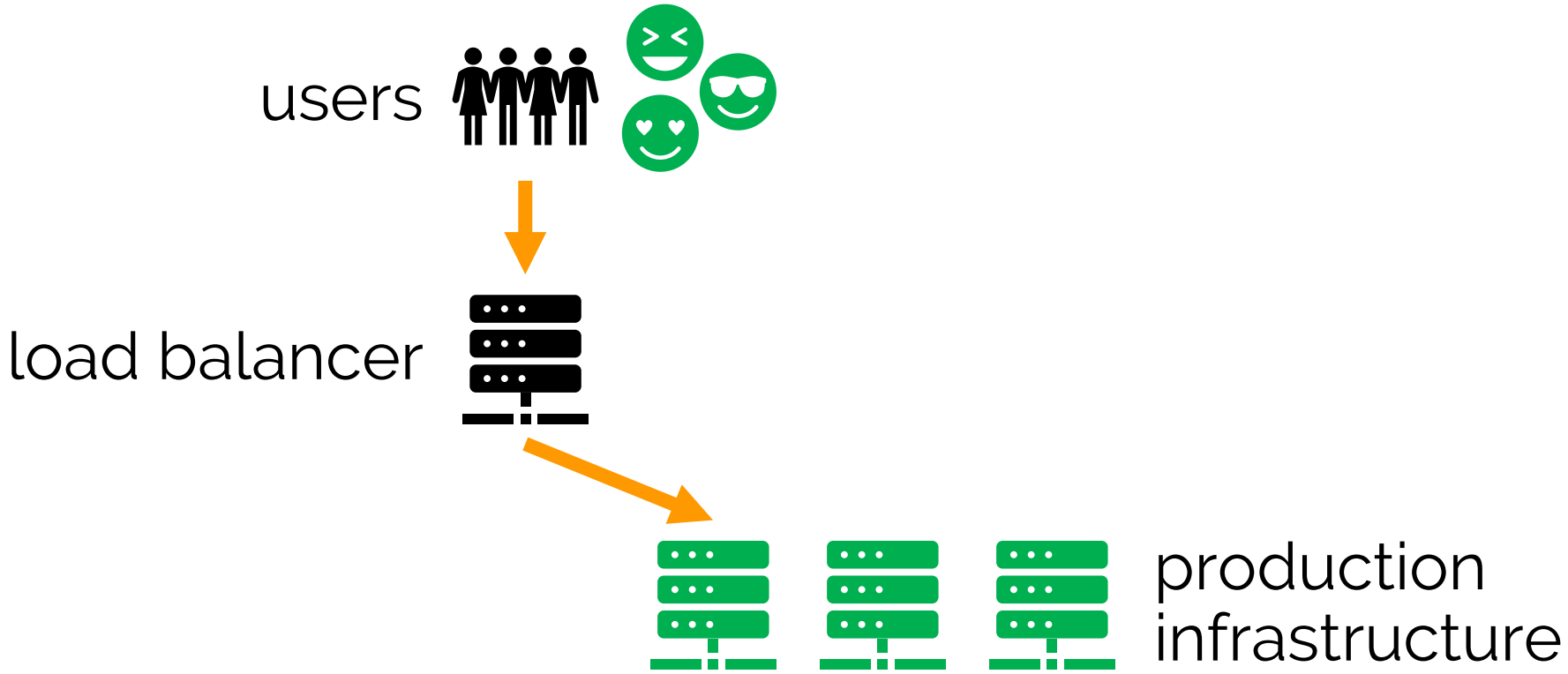
- Blue – Green deployments
- Canary
- Cluster immune system

Application-based

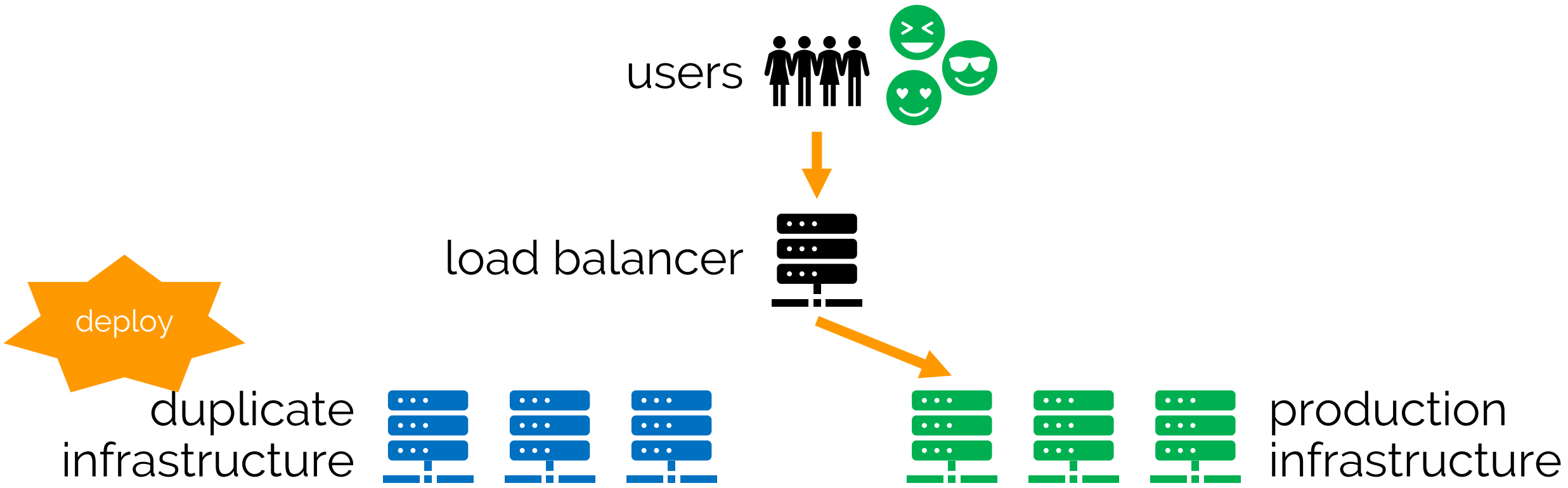
- Feature toggles
- Dark launches

Environment-based

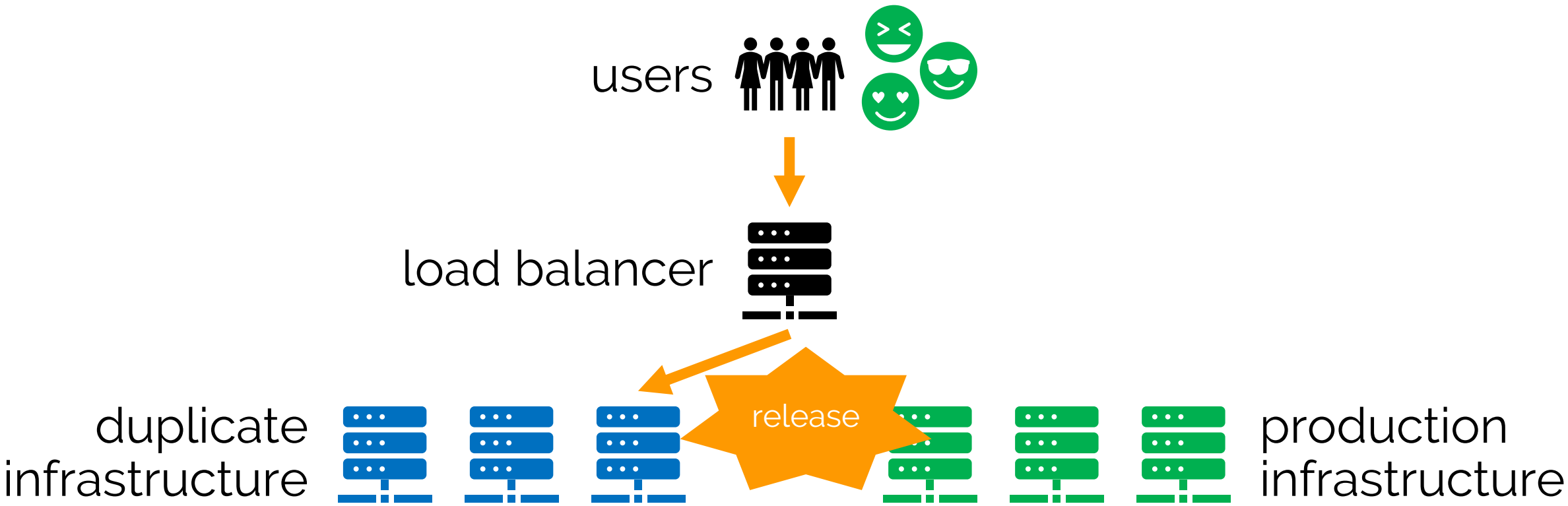
Blue – Green deployments



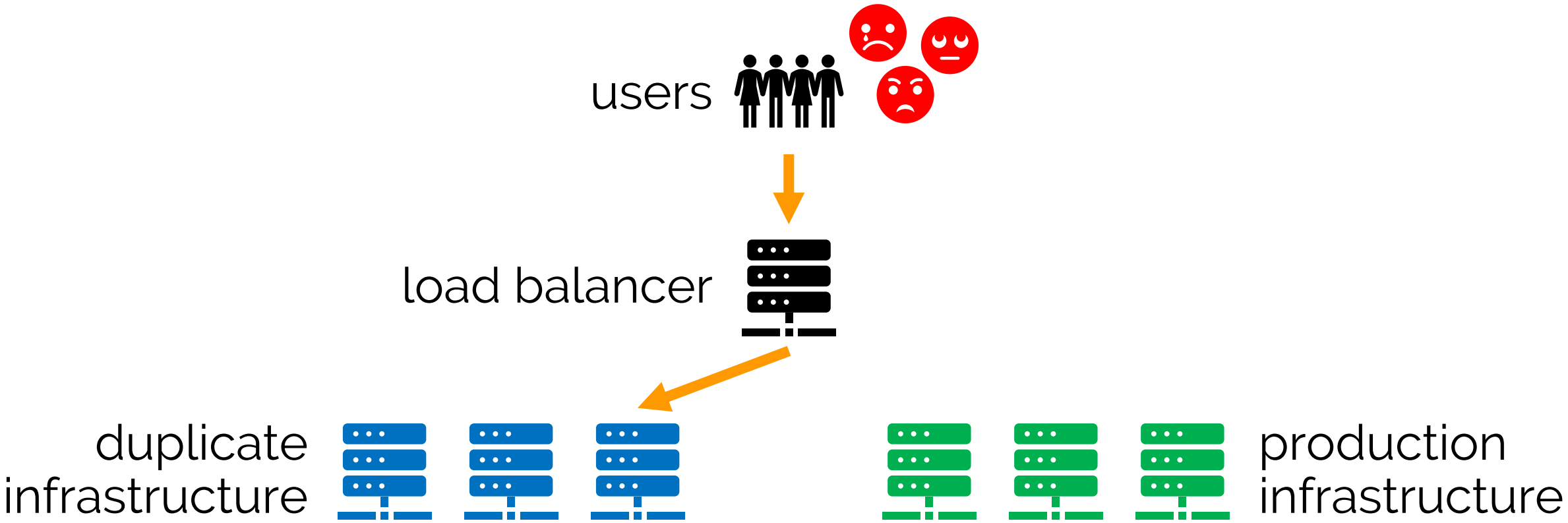
Blue – Green deployments



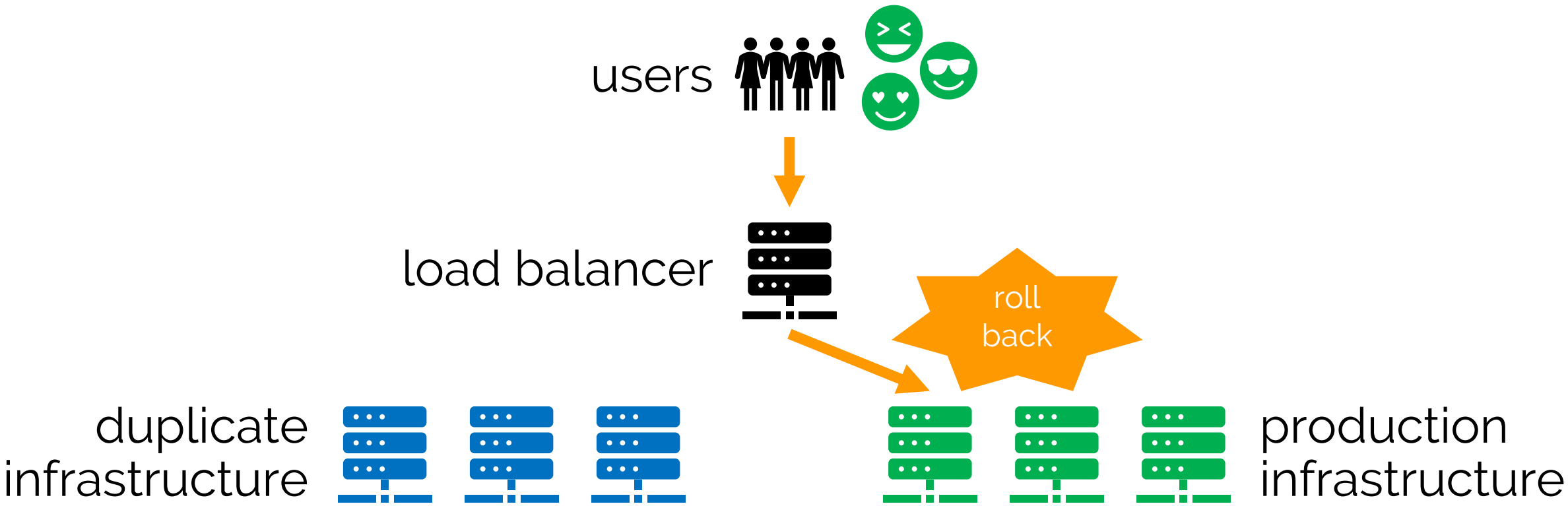
Blue – Green deployments



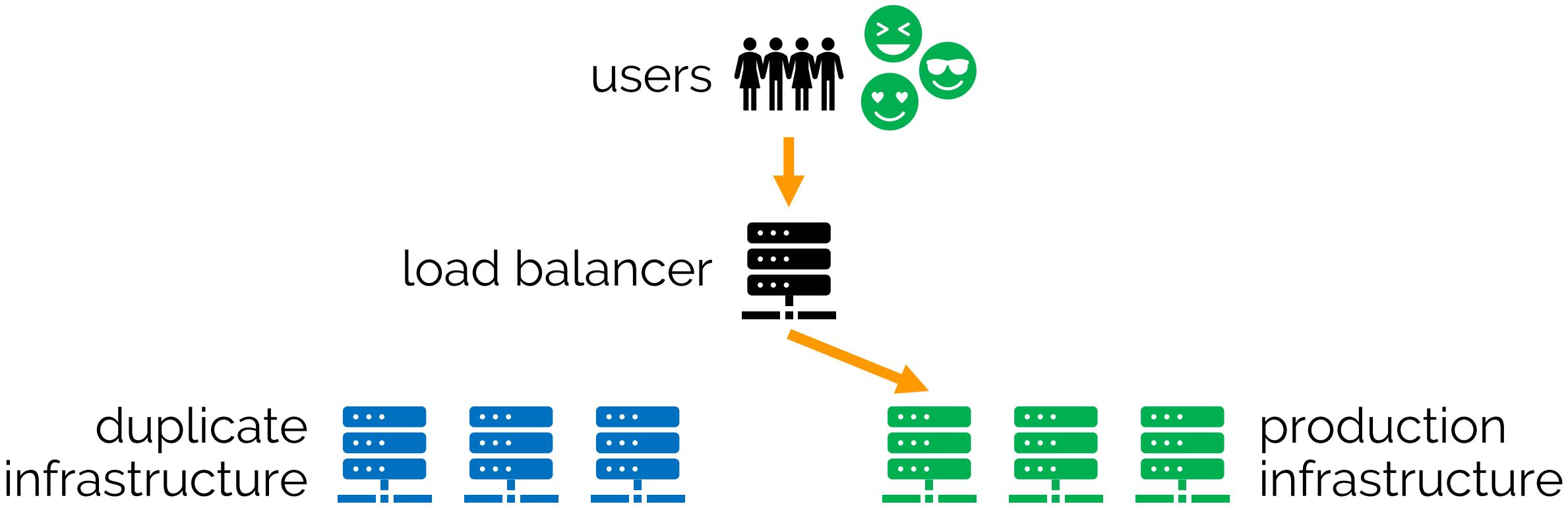
Blue – Green deployments



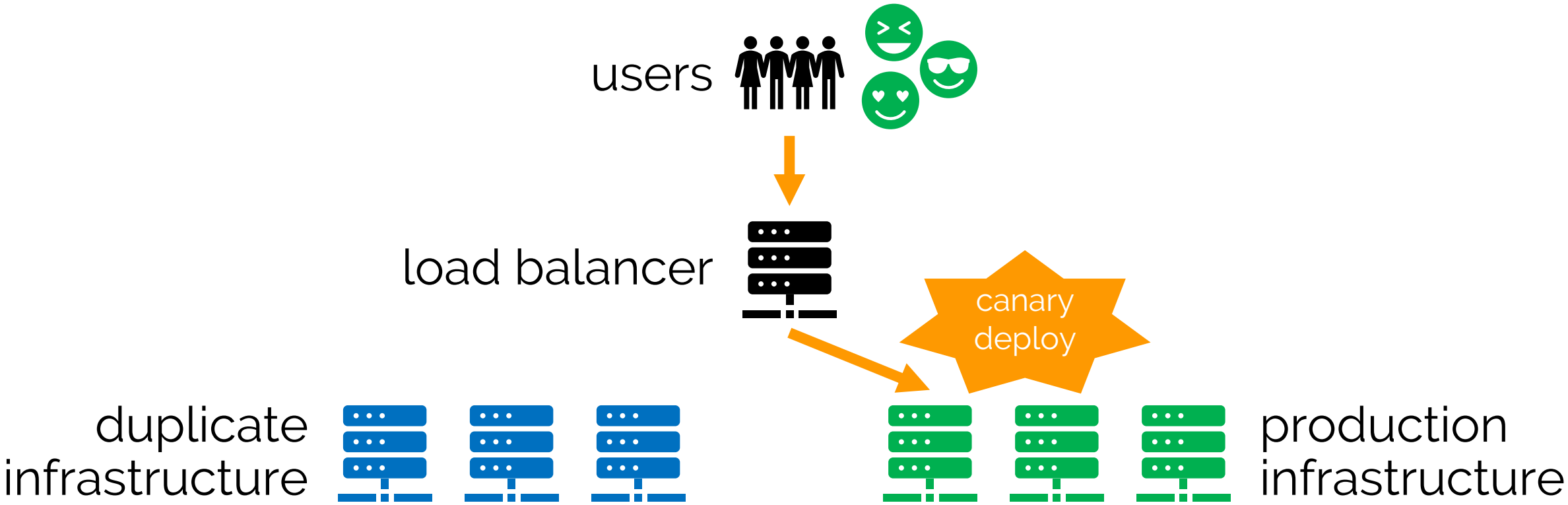
Blue – Green deployments



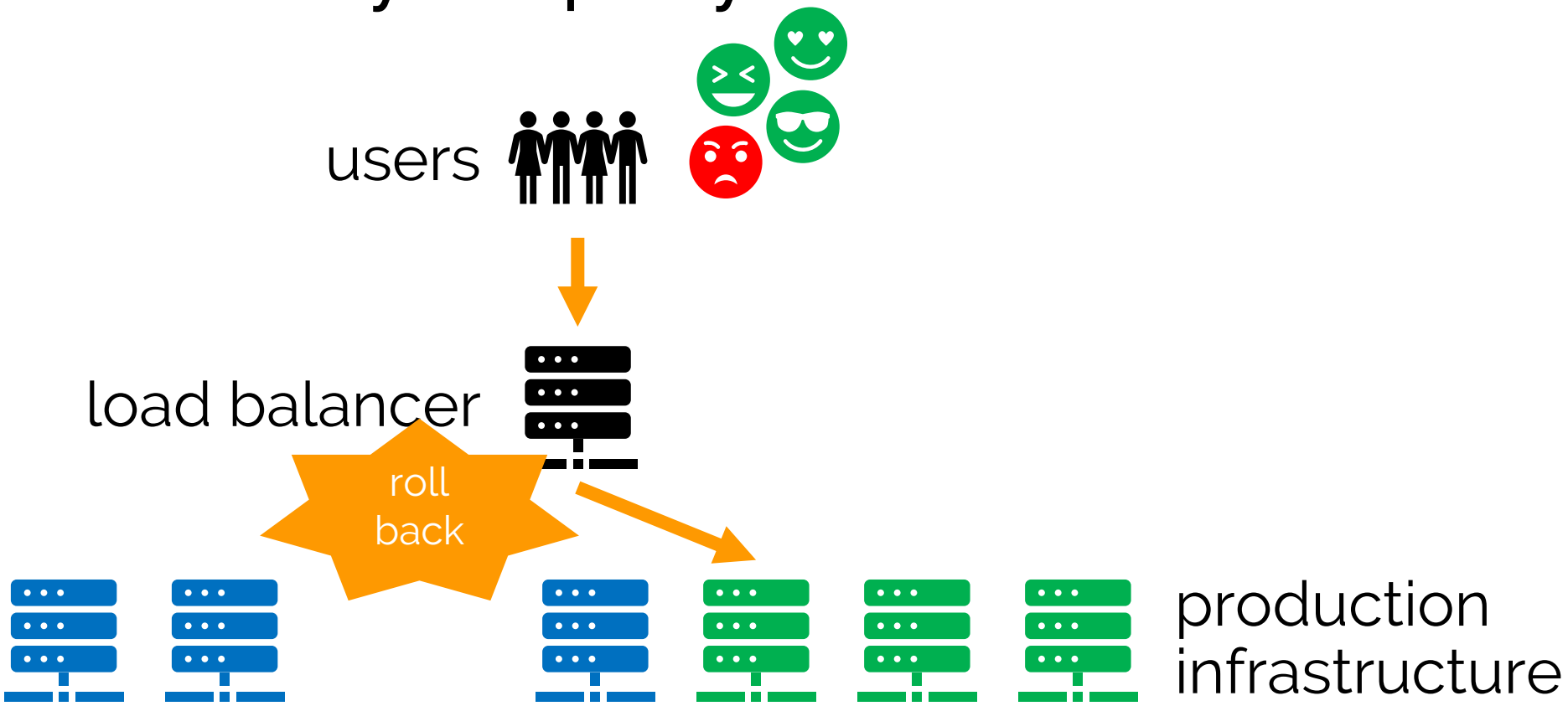
Canary deployments



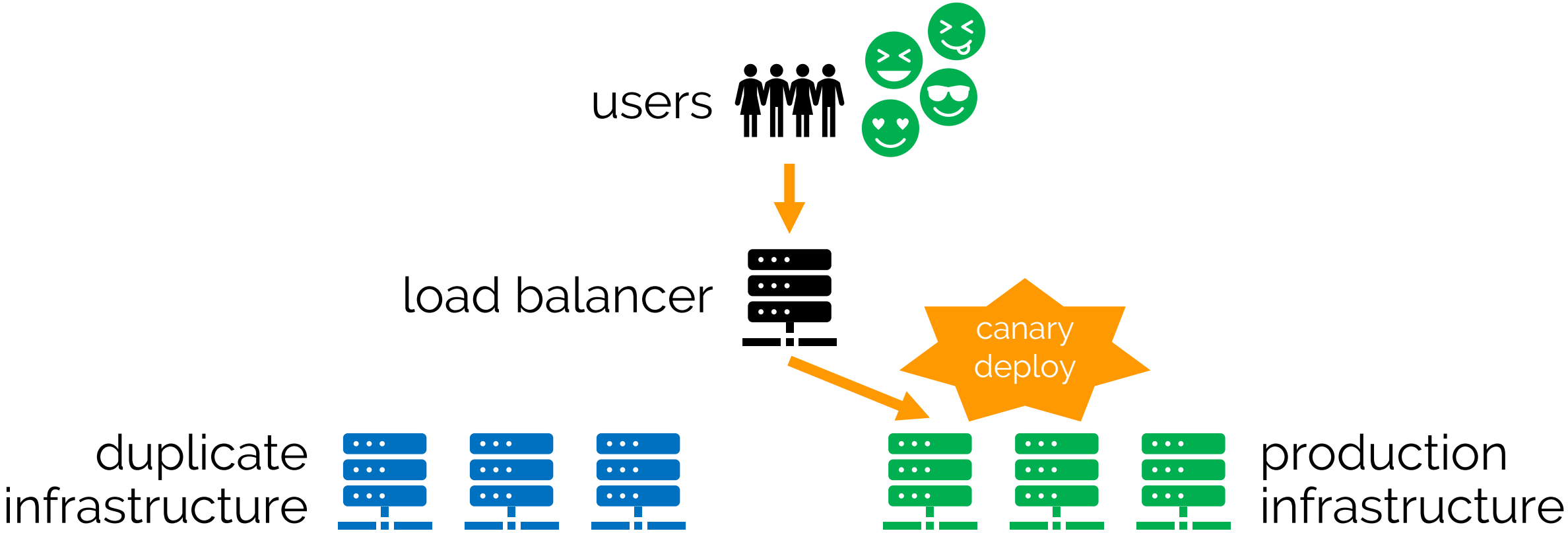
Canary deployments



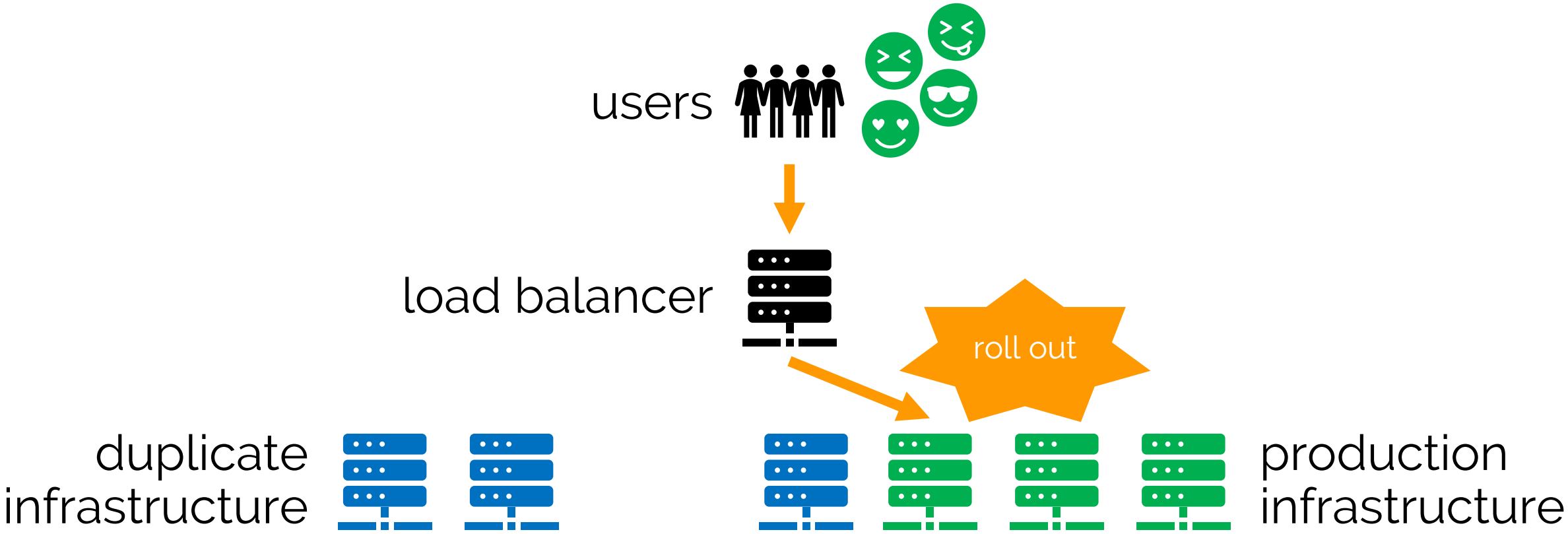
Canary deployments



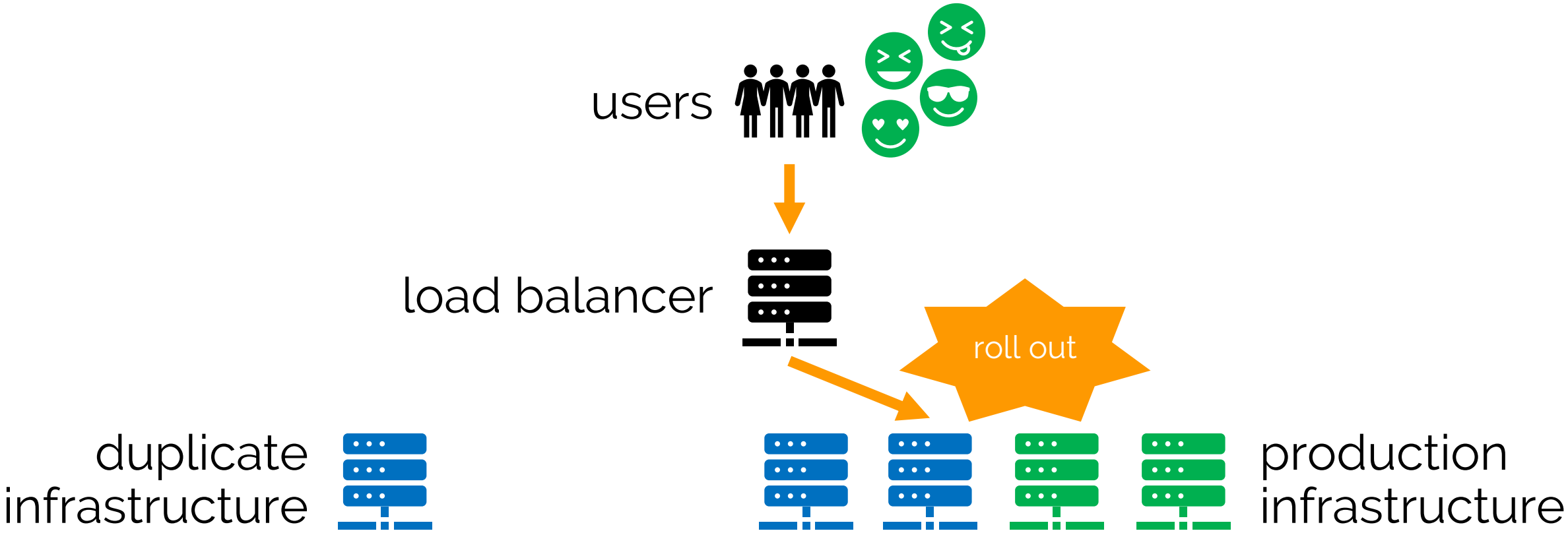
Canary deployments



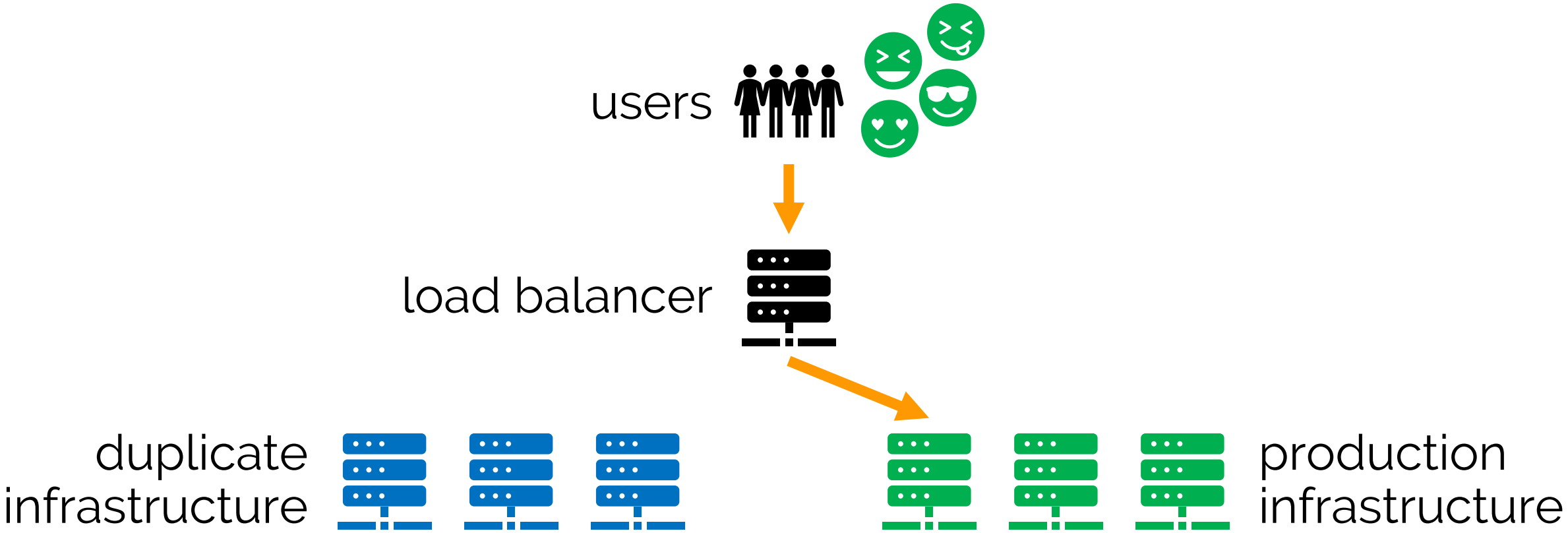
Canary deployments



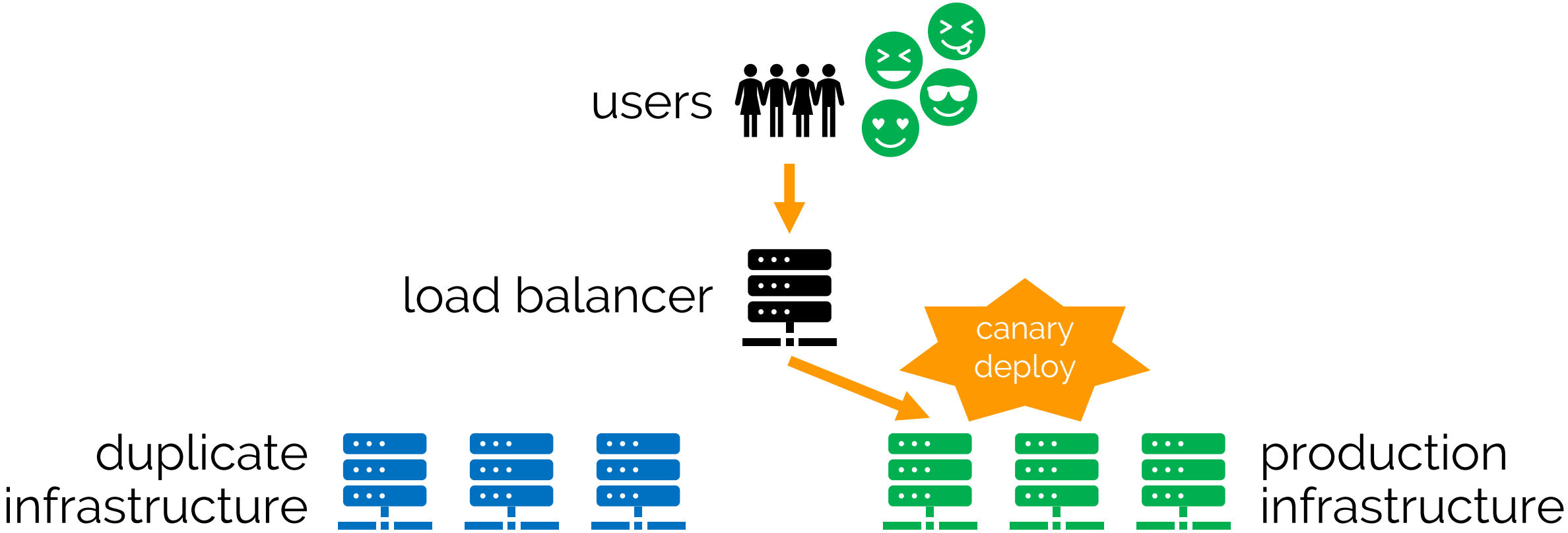
Canary deployments



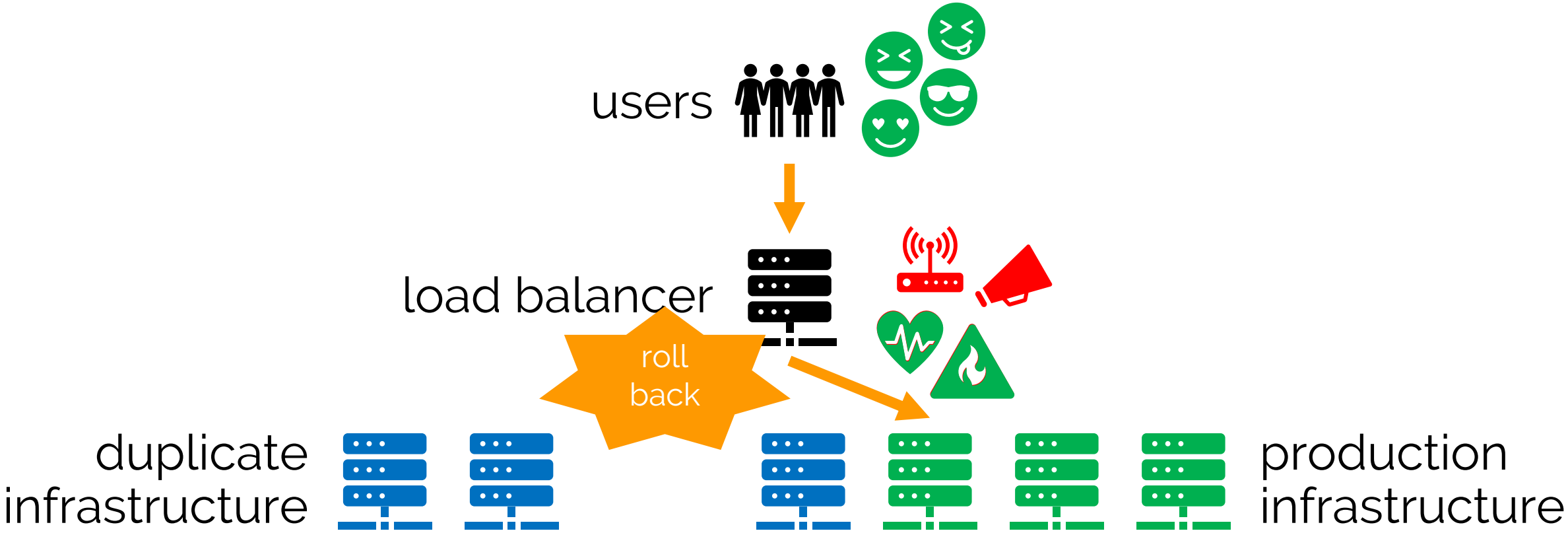
Cluster immune systems



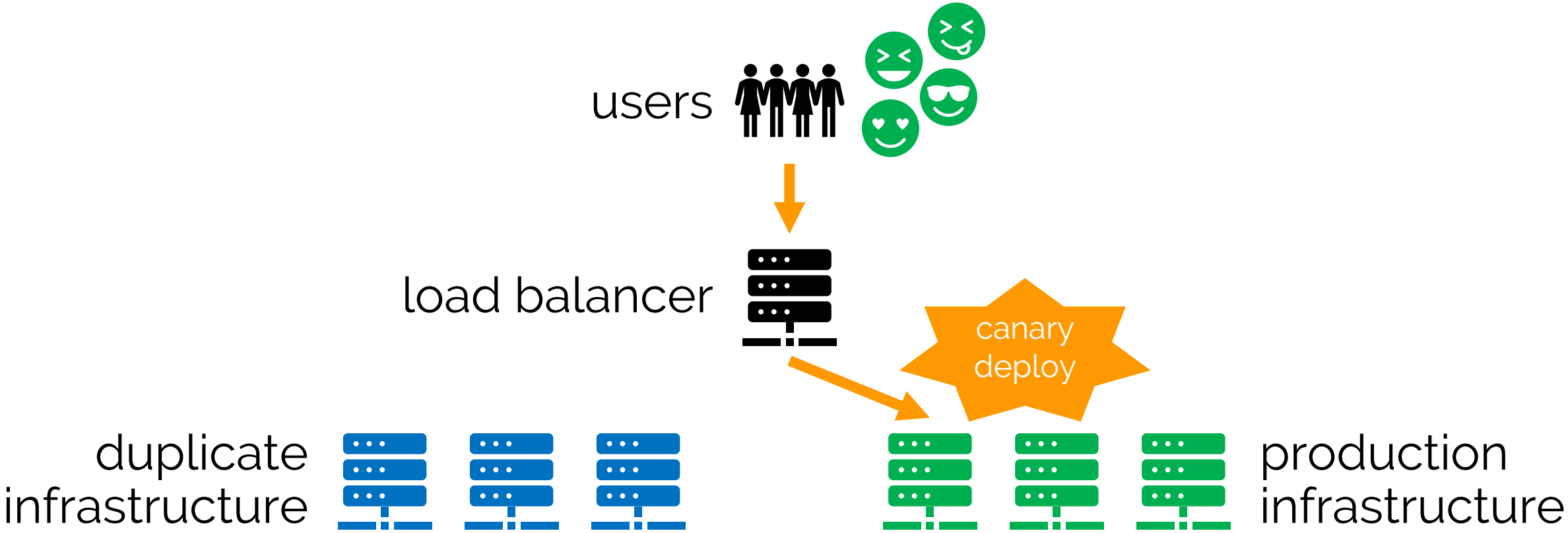
Cluster immune systems



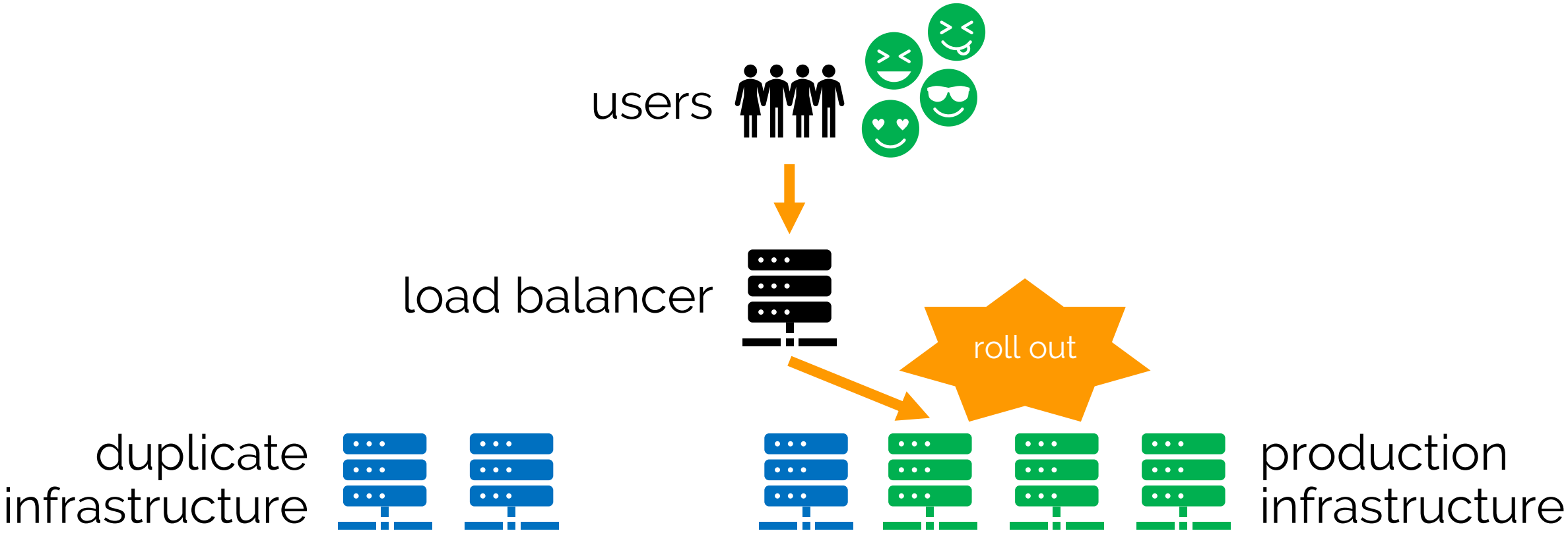
Cluster immune systems



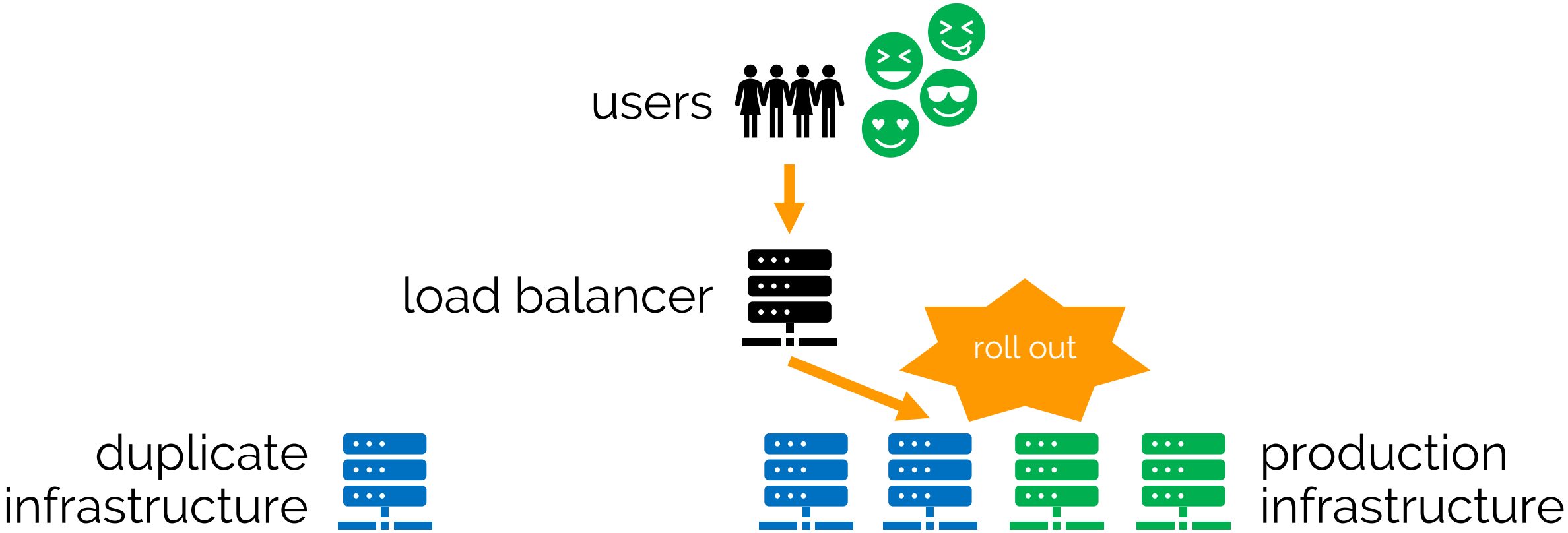
Cluster immune systems



Cluster immune systems



Cluster immune systems



Application-based

Feature Toggles

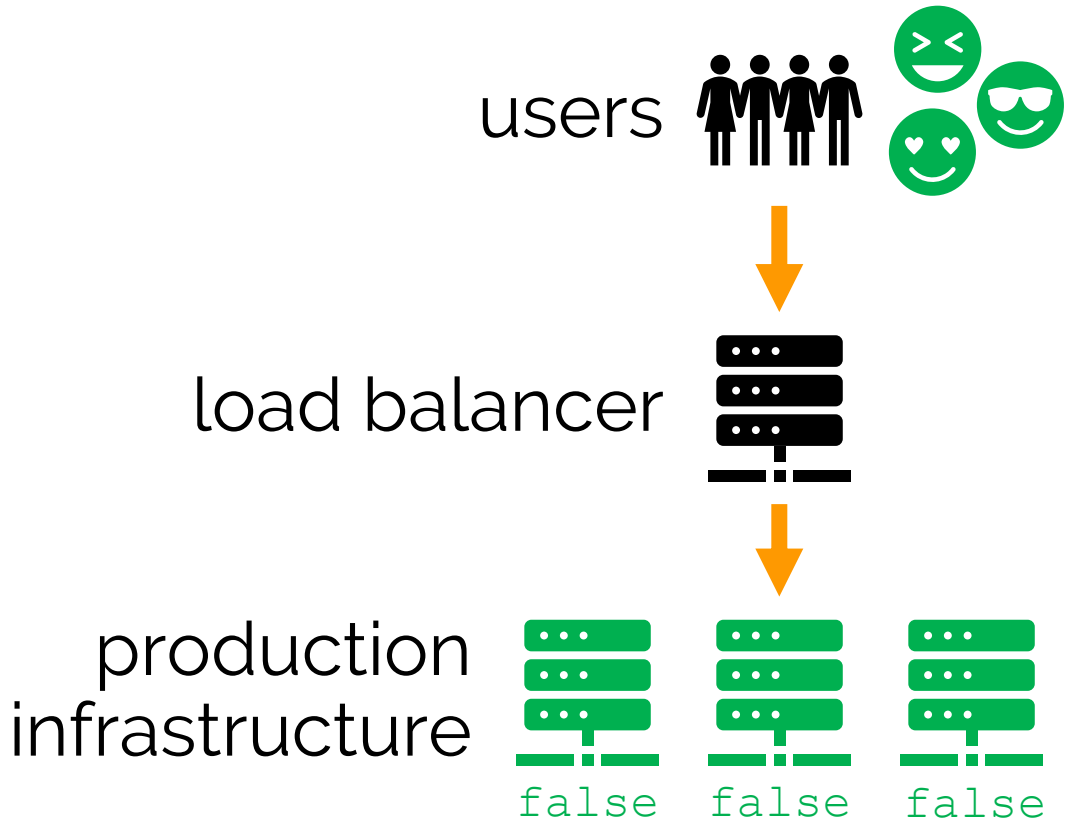
```
function reticulateSplines(){
  // current implementation lives here
}
```

```
// Attribution: https://martinfowler.com/articles/feature-toggles.html
```

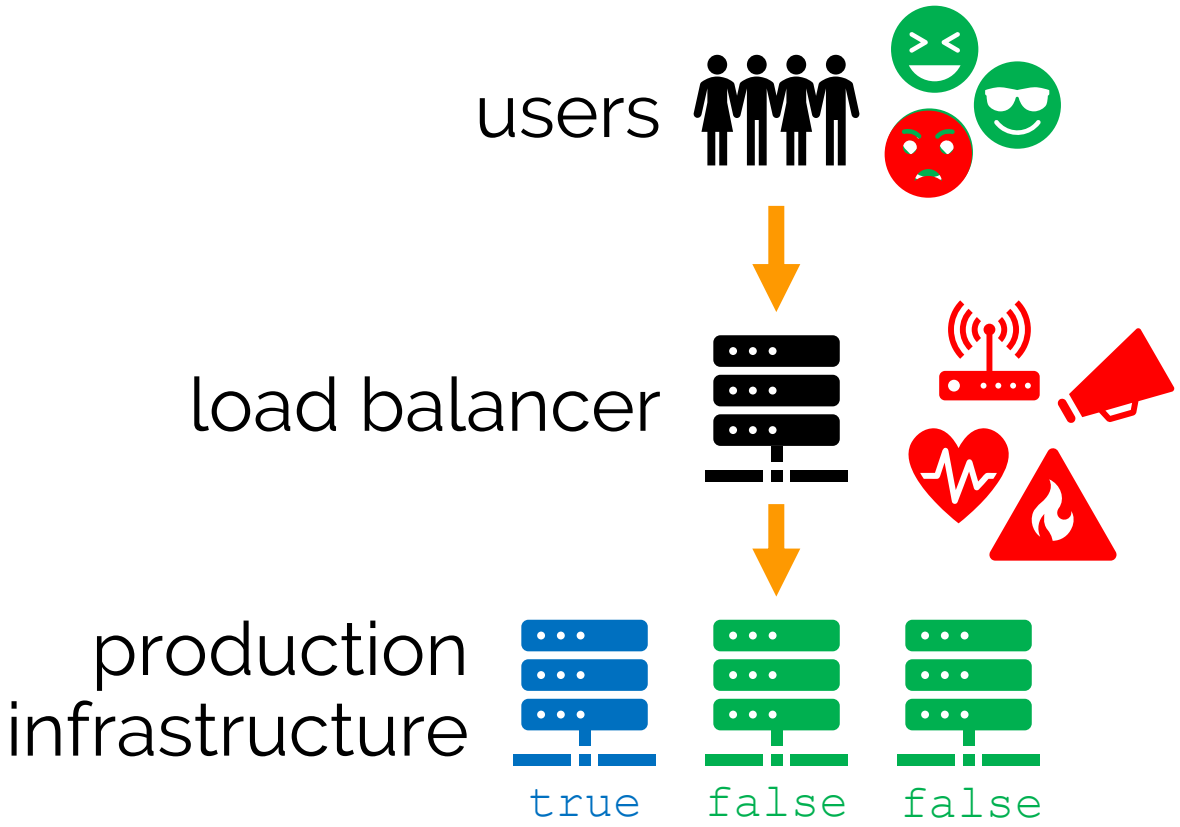
Feature Toggles

```
function reticulateSplines(){
  var useNewAlgorithm = false;
  // useNewAlgorithm = true; // UNCOMMENT IF YOU ARE WORKING ON THE NEW SR ALGORITHM

  if( useNewAlgorithm ){
    return enhancedSplineReticulation();
  }else{
    return oldFashionedSplineReticulation();
  }
}

function oldFashionedSplineReticulation(){
  // current implementation lives here
}

function enhancedSplineReticulation(){
  // TODO: implement better SR algorithm
}

// Attribution: https://martinfowler.com/articles/feature-toggles.html
```

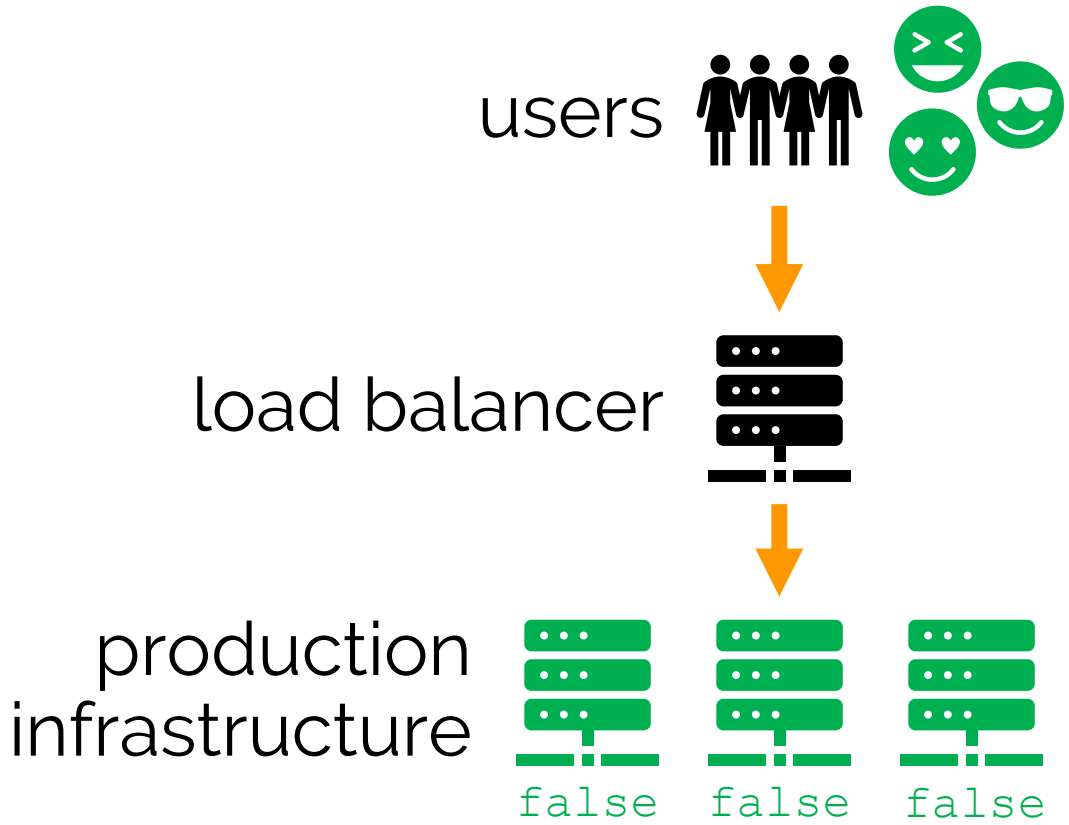
Feature Toggles



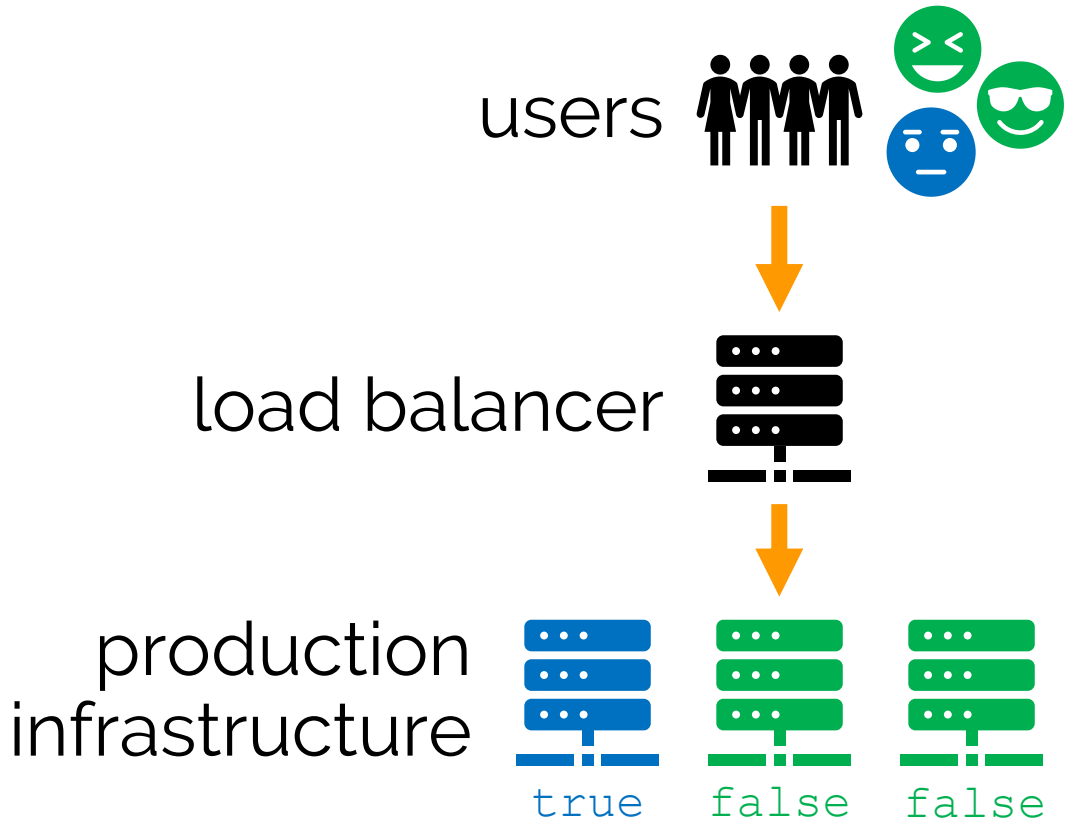
Feature Toggles



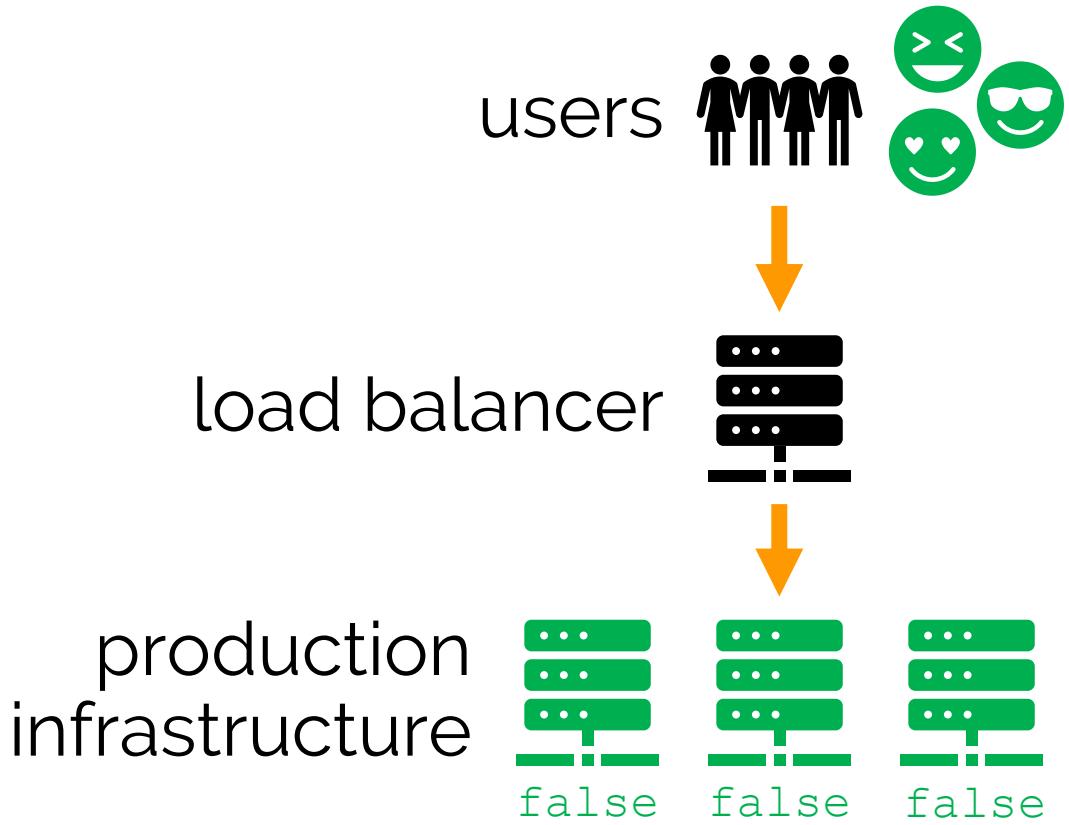
Feature Toggles



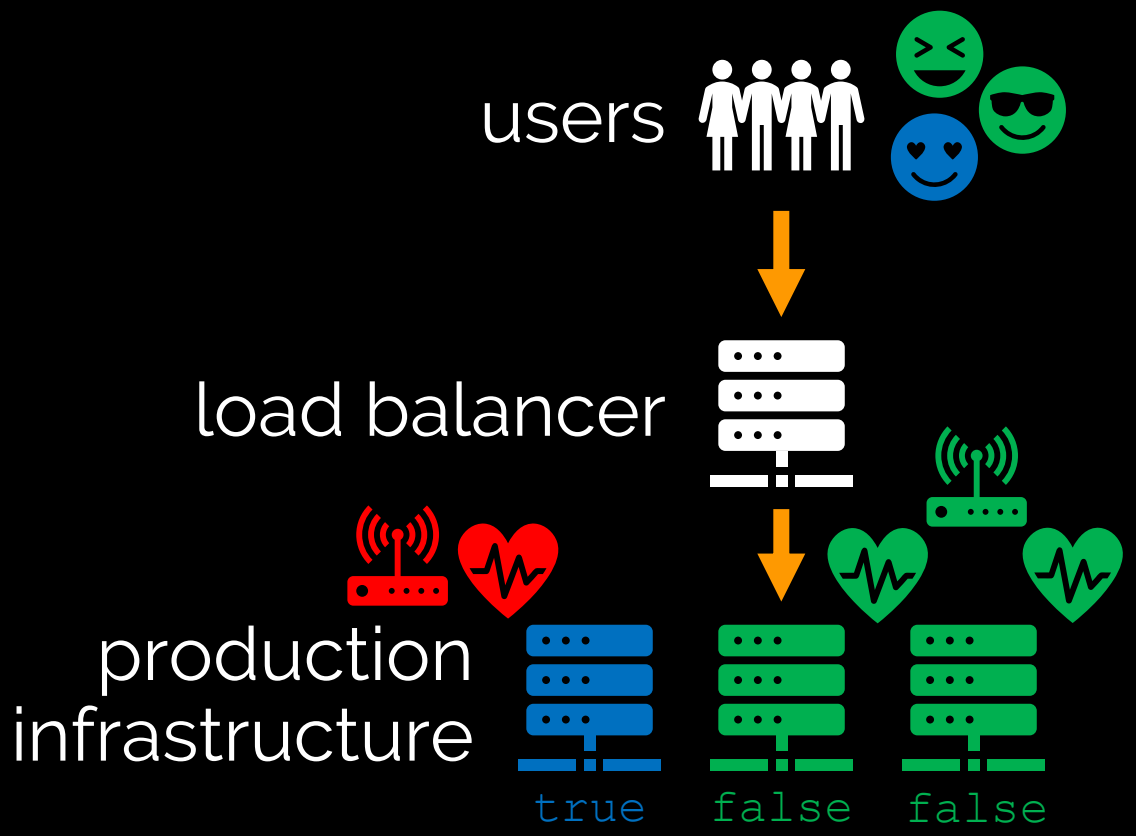
Feature Toggles



Feature Toggles



Dark launches



Release patterns

Environment-based

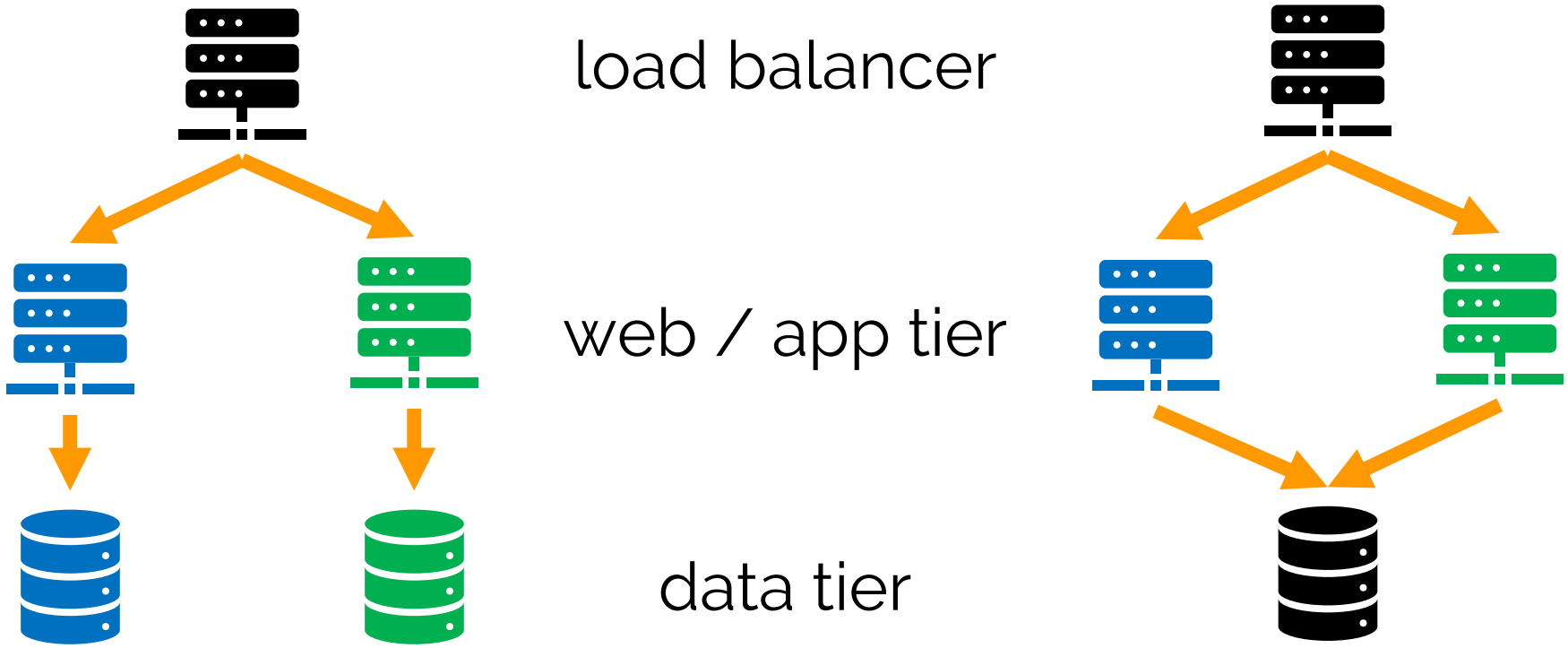
- Blue – Green deployments
- Canary
- Cluster immune system

Application-based

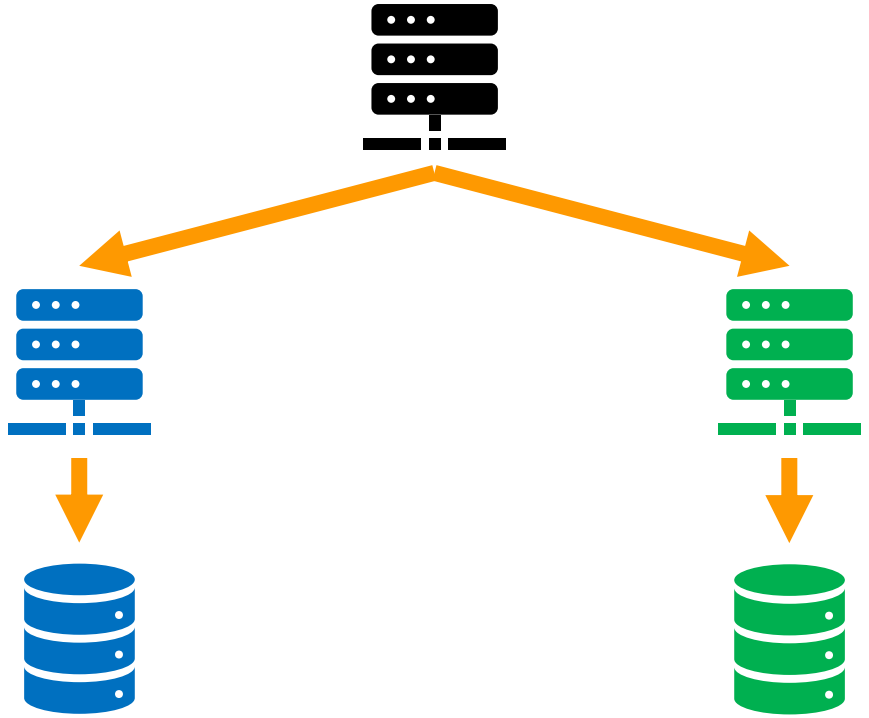
- Feature toggles
- Dark launches

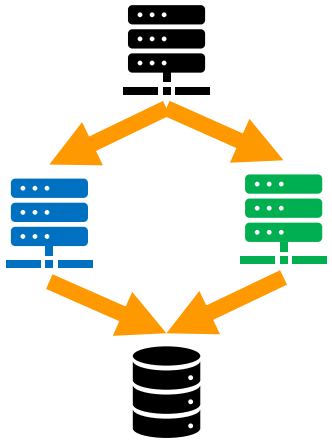
Let's talk about data

Data and databases

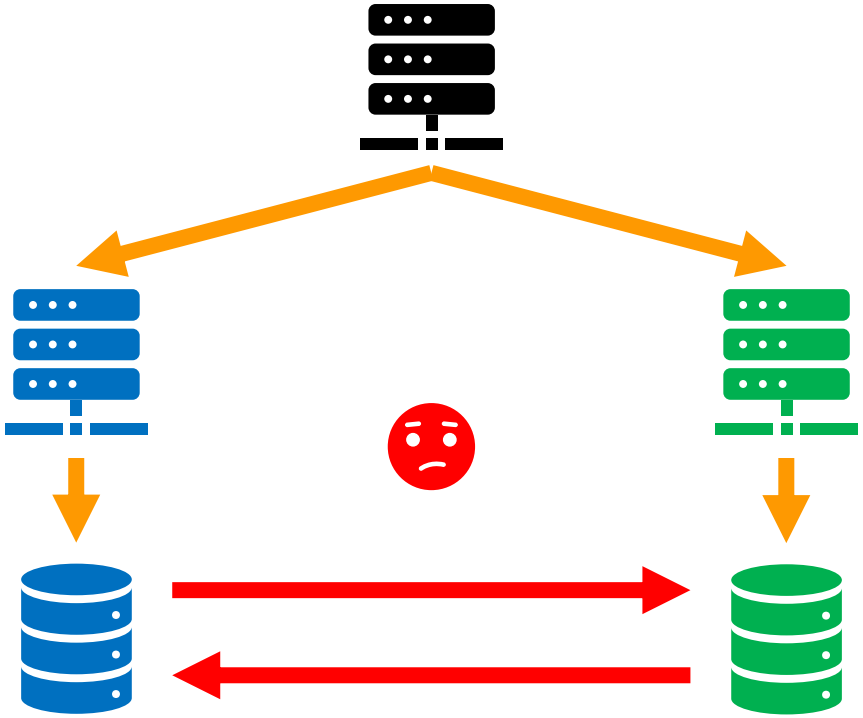


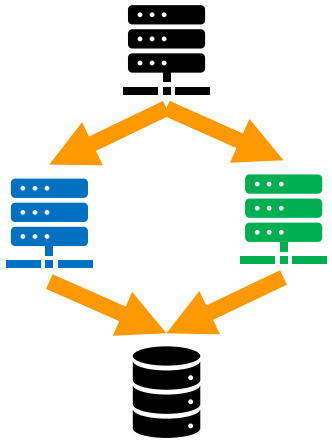
Data and databases



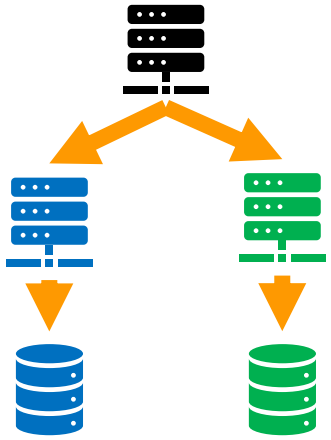


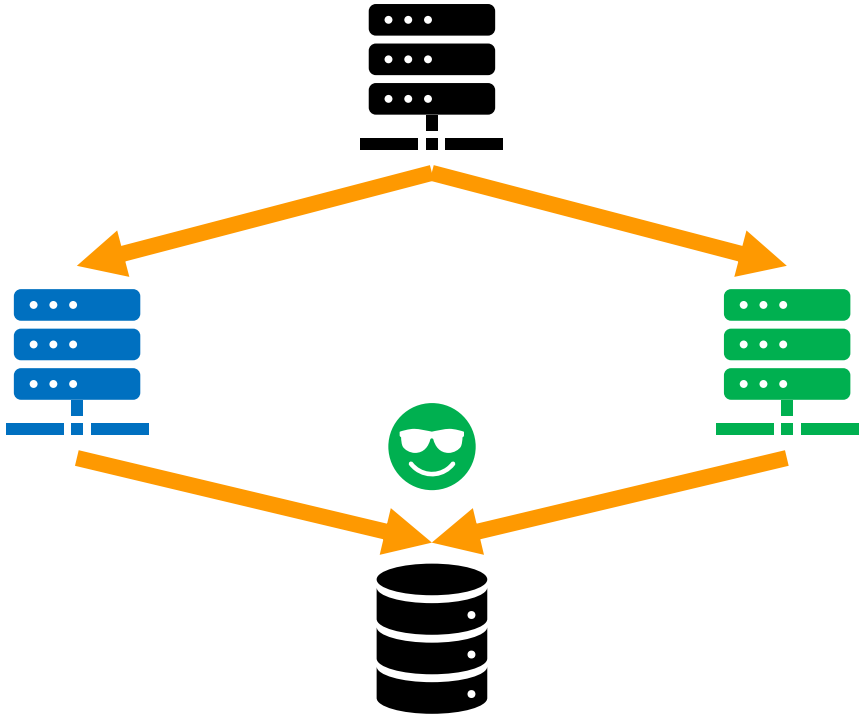
Data and databases





Data and databases





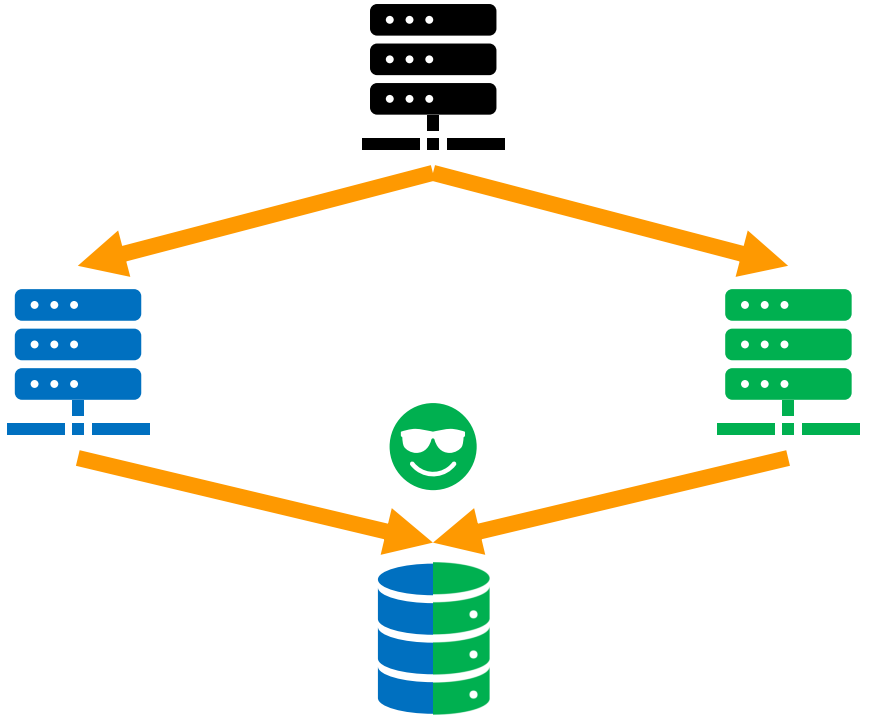
@_AlexYates_

#DevOps

Zero Downtime Database Deployments

DLM Consultants

Data and databases



@_AlexYates_

#DevOps

Zero Downtime Database Deployments

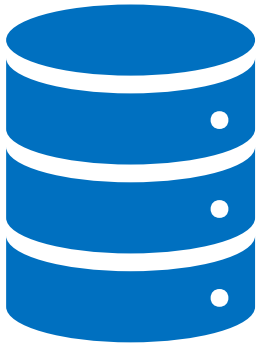
DLM Consultants

Expand / Contract

Splitting a column

FullName
Martin Fowler
Pete Hodgeson
Danilo Sato

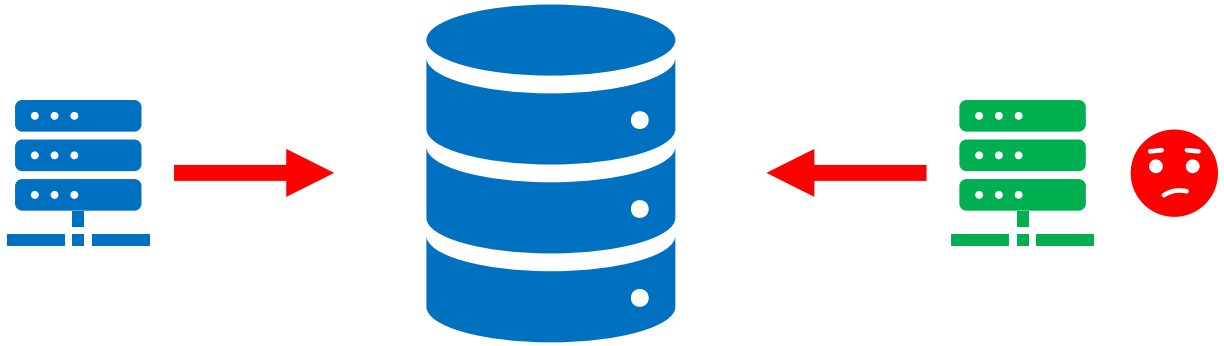
Splitting a column



FirstName	LastName
Martin	Fowler
Pete	Hodgeson
Danilo	Sato

FullName
Martin Fowler
Pete Hodgeson
Danilo Sato

Splitting a column



FirstName	LastName
Martin	Fowler
Pete	Hodgeson
Danilo	Sato

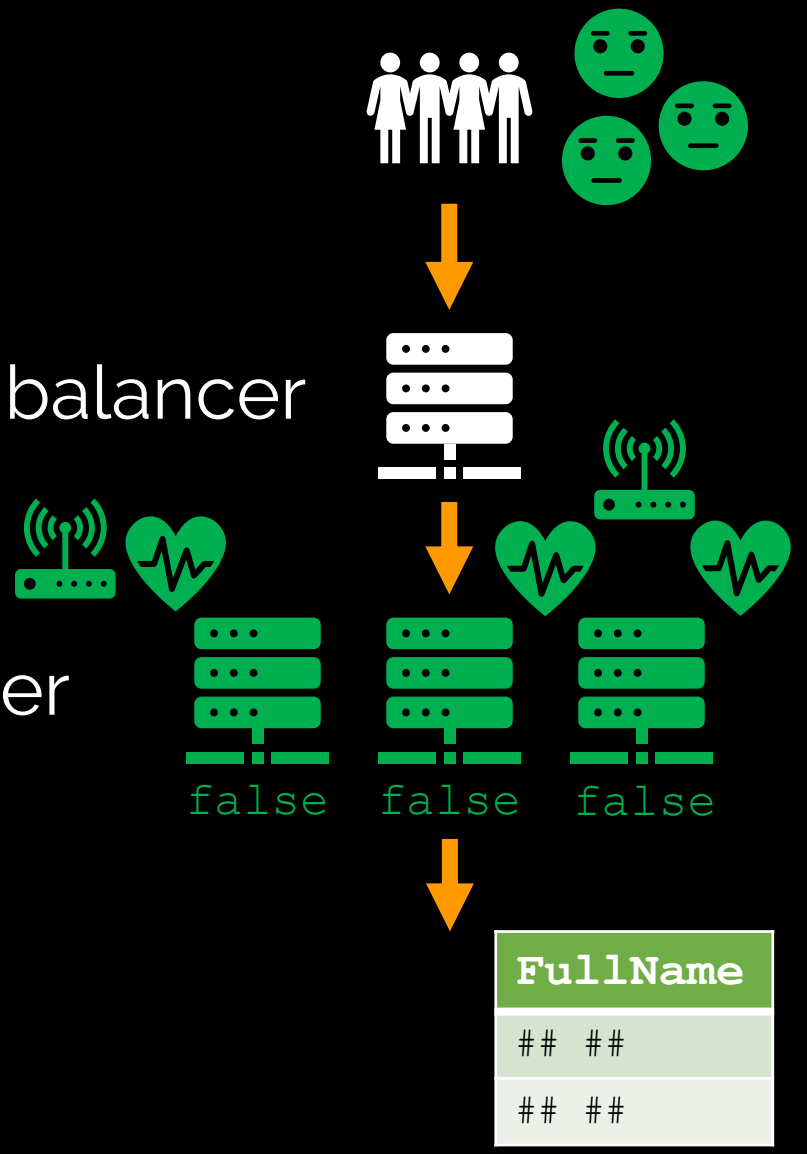
Splitting a column

FullName
Martin Fowler
Pete Hodgeson
Danilo Sato

Splitting a column

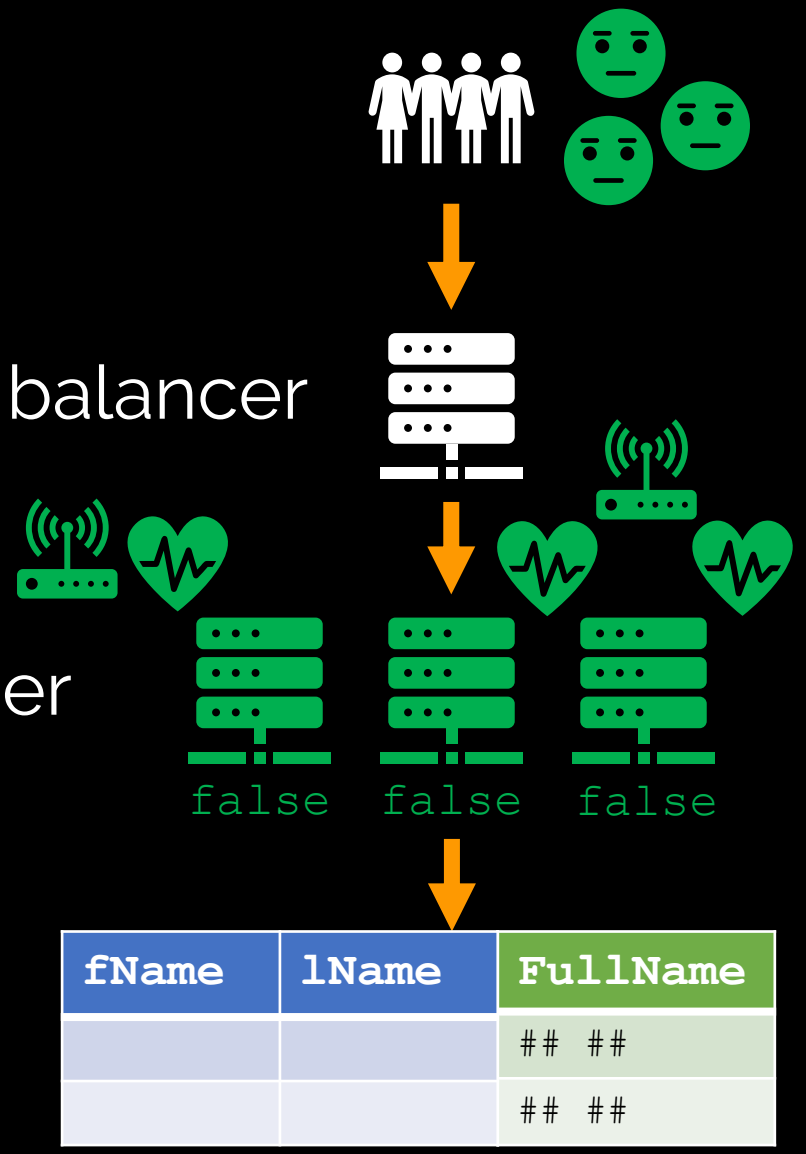
FirstName	LastName	FullName
Martin	Fowler	Martin Fowler
Pete	Hodgeson	Pete Hodgeson
Danilo	Sato	Danilo Sato

Putting it all together



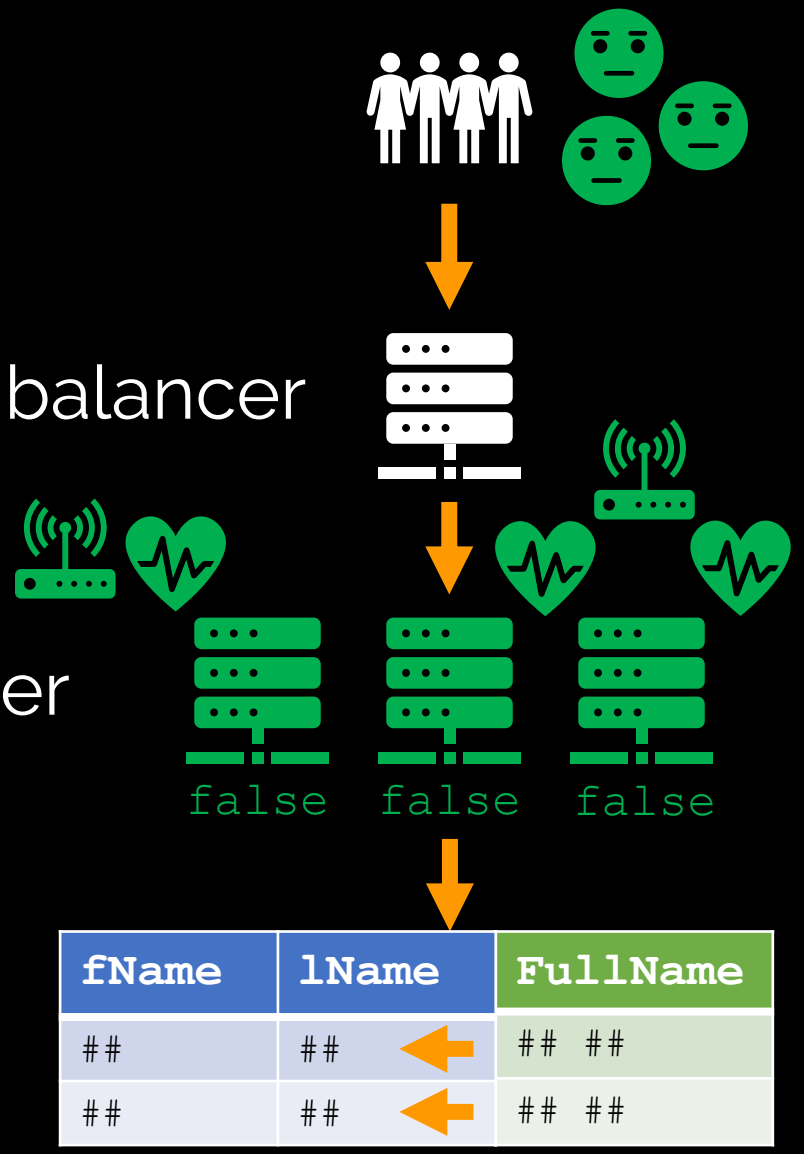
Putting it all together

1. Expand DB



Putting it all together

1. Expand DB



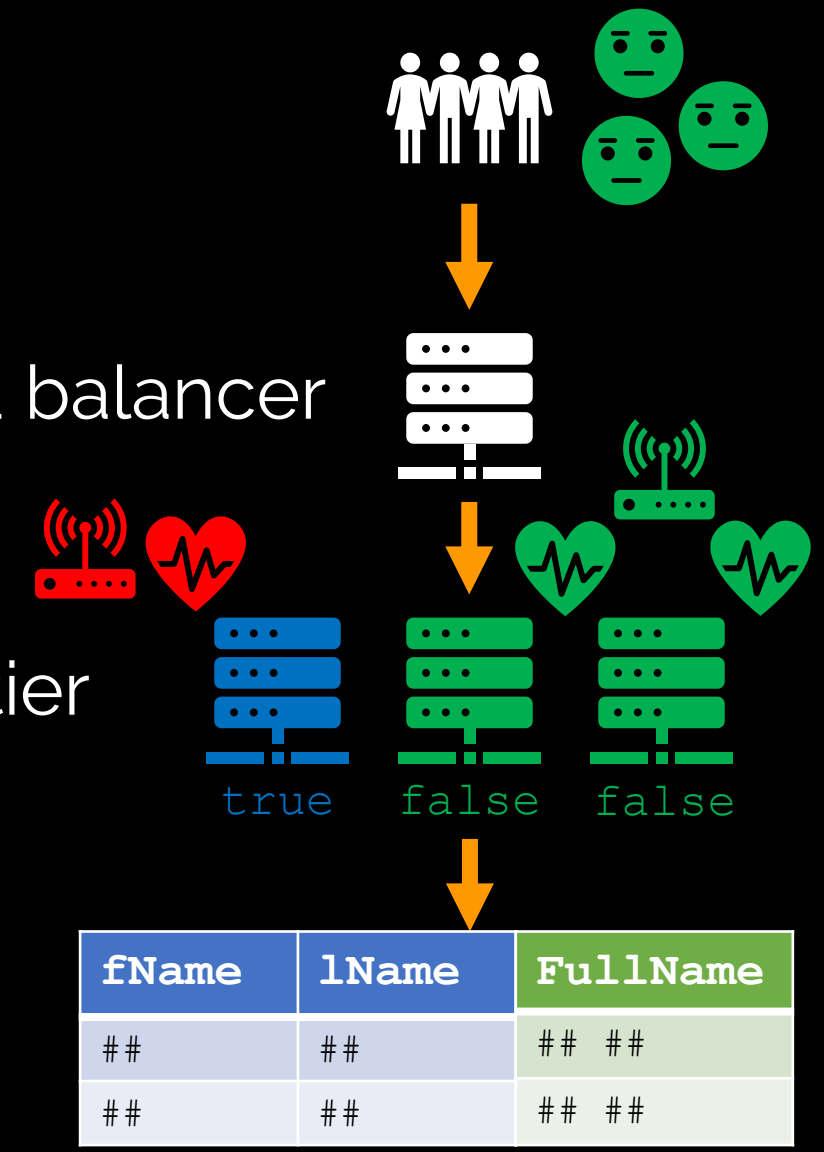
@_AlexYates_
#DevOps

Zero Downtime Database Deployments

DLM Consultants

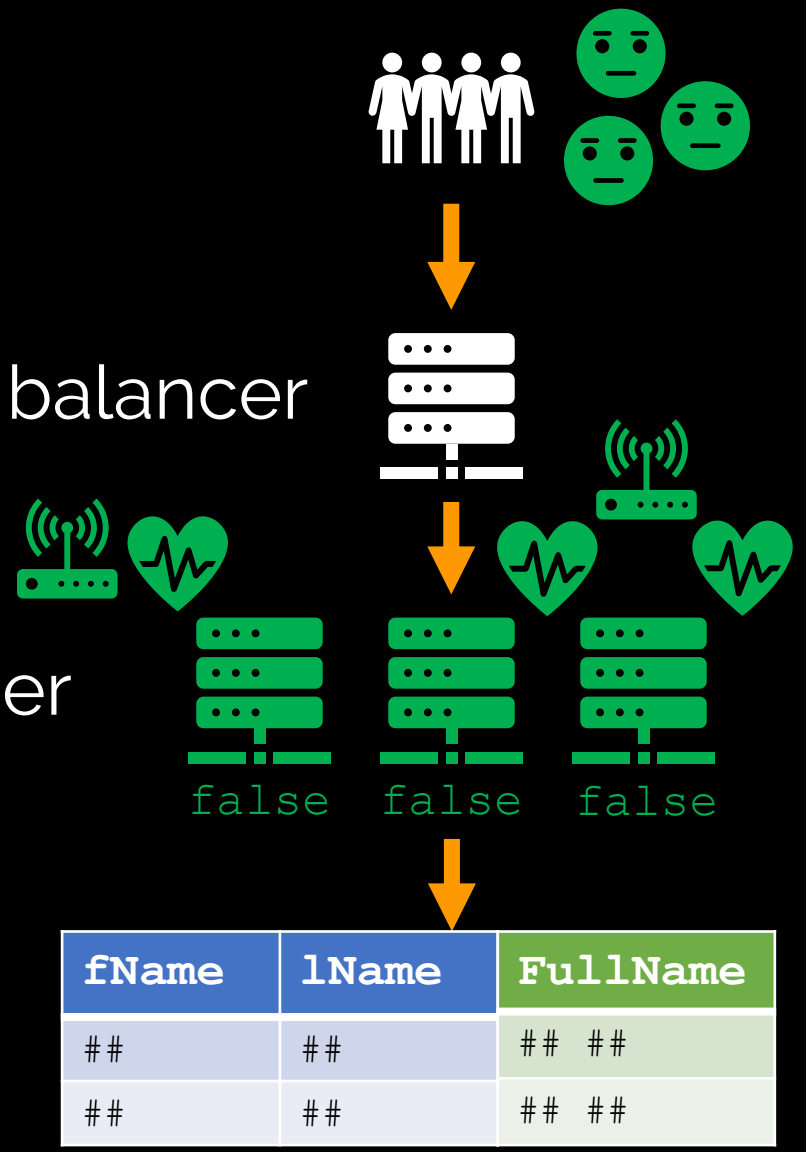
Putting it all together

1. Expand DB
2. Release canary



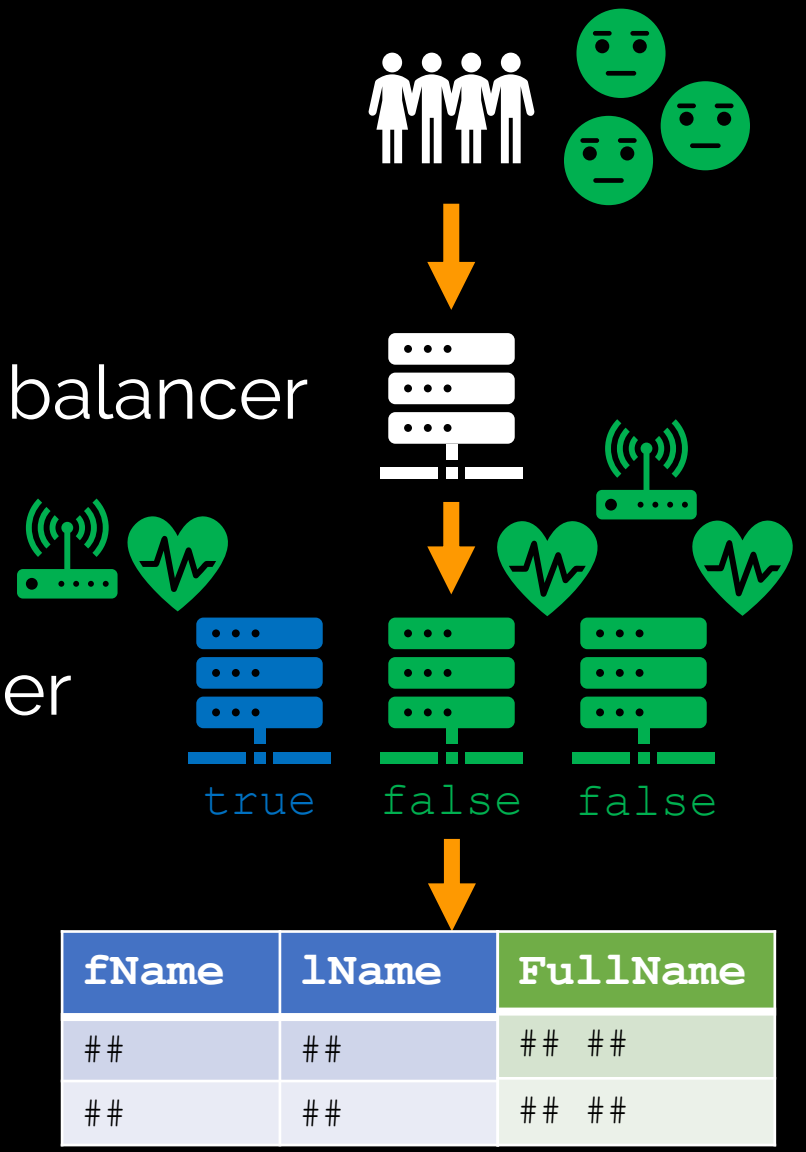
Putting it all together

1. Expand DB
2. Release canary



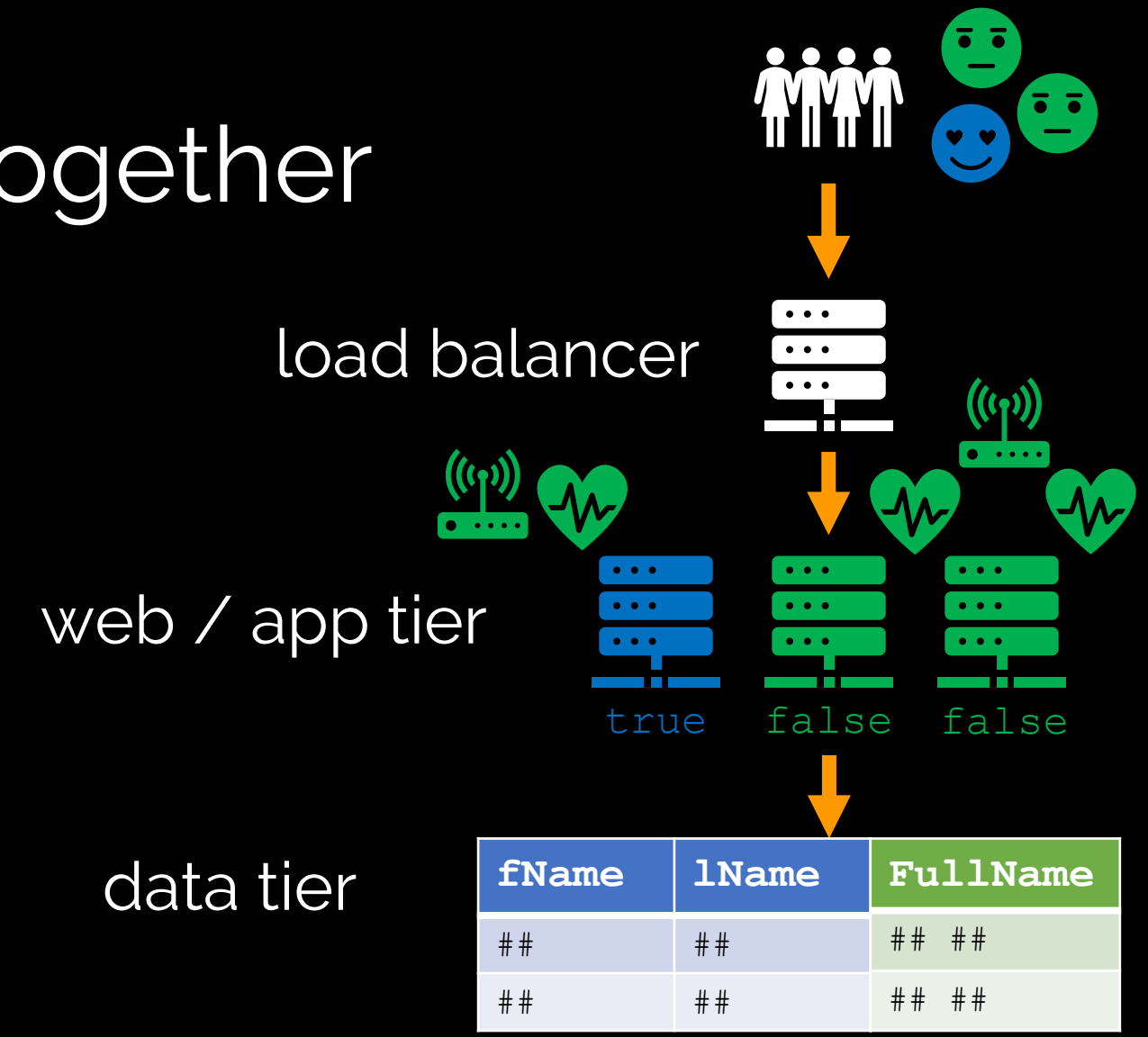
Putting it all together

1. Expand DB
2. Release canary



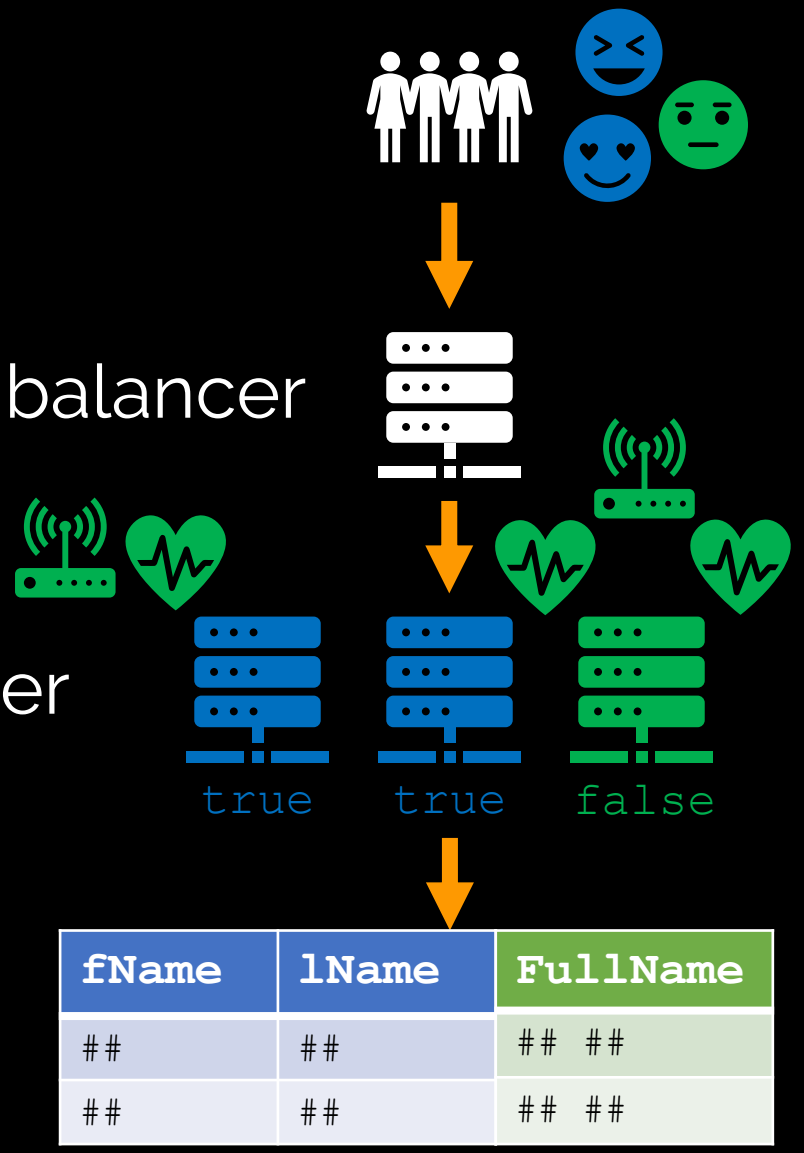
Putting it all together

1. Expand DB
2. Release canary
3. Check telemetry



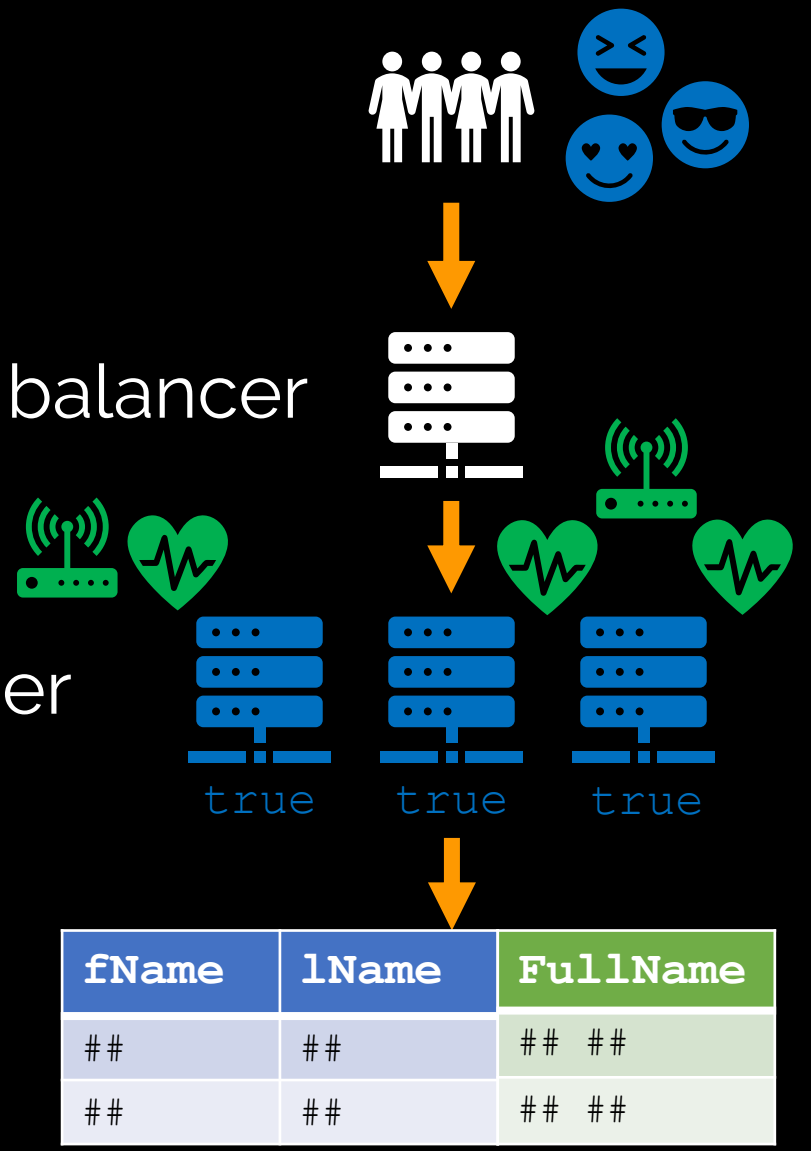
Putting it all together

1. Expand DB
2. Release canary
3. Check telemetry
4. Roll out



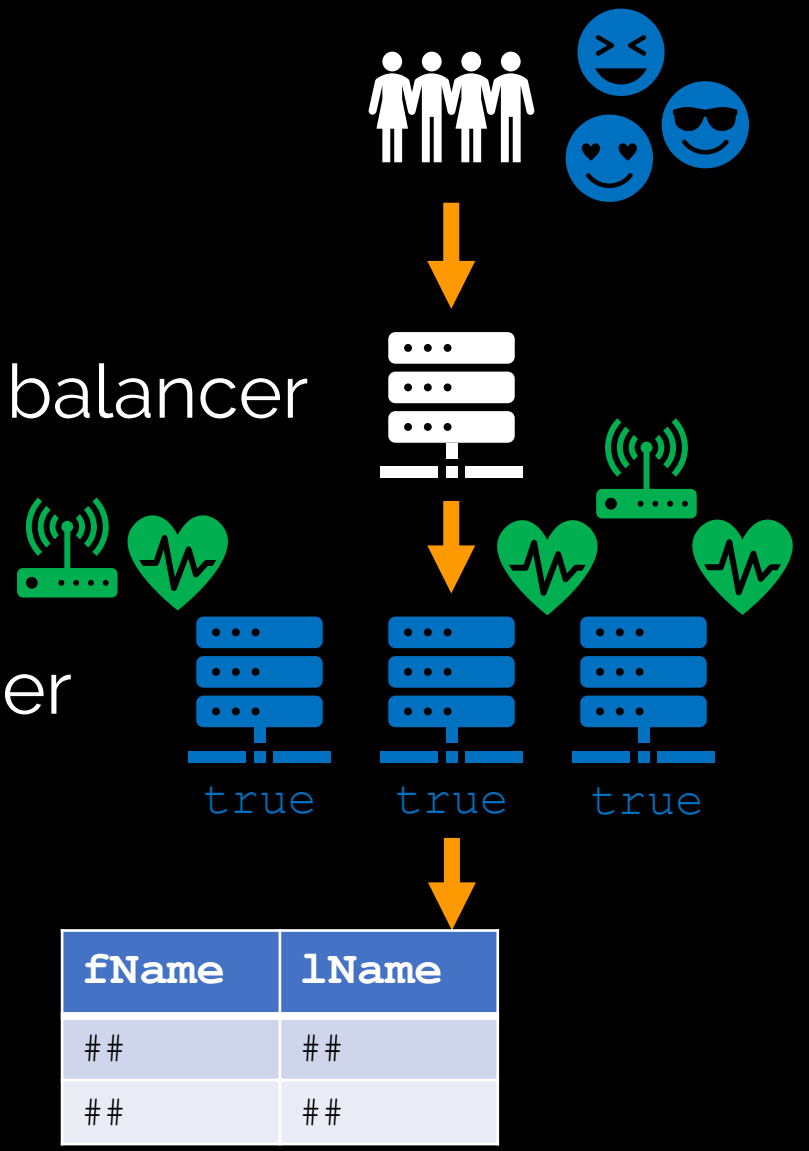
Putting it all together

1. Expand DB
2. Release canary
3. Check telemetry
4. Roll out



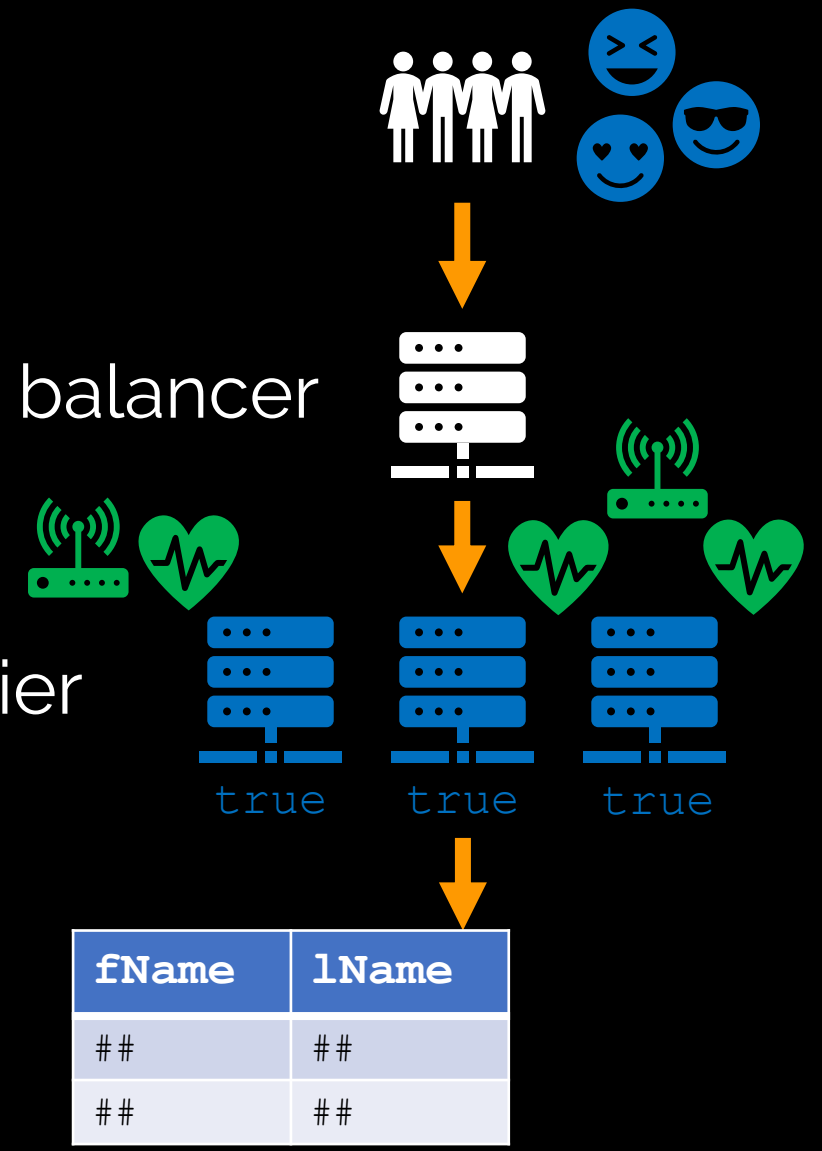
Putting it all together

1. Expand DB
2. Release canary
3. Check telemetry
4. Roll out
5. Contract DB



Putting it all together

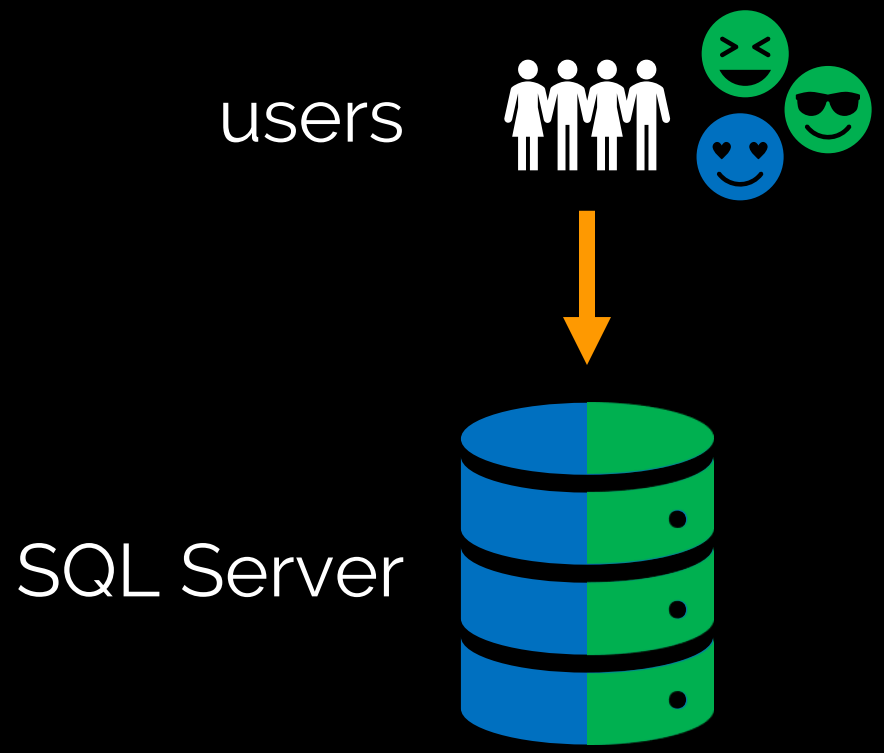
1. Expand DB
2. Release canary
3. Check telemetry
4. Roll out
5. Contract DB (Rename first or hide behind view/sproc)



Zero Downtime Database Deployments

DLM Consultants

Dark launches and Feature Toggles:
SQL Server Demo!



Further reading...

Deploy != Release (Part 1)

The difference between deploy and release and why it matters.

Art Gillespie Follow
May 24, 2017 · 5 min read

Q: "Is the latest version deployed?"

A: "I deployed animated gif support to production."

Q: "So animated gif support is released?"

A: "The animated gif release is deployed."

Q: "…"

I've worked at many companies where "deploy", "deployment", "ship", and "release" are used loosely, even interchangeably. As an industry, we haven't done a great job of standardizing our use of these terms, even though we've radically improved operations practices and tooling over the past decade. At Turbine Labs, we use precise definitions of "ship", "deploy", "release", and

blog.turbinelabs.io/deploy-not-equal-release-part-one-4724bc1e726b

martinFowler.com

Refactoring Agile Architecture About ThoughtWorks

Feature Toggles (aka Feature Flags)

Feature Toggles (often also refered to as Feature Flags) are a powerful technique, allowing teams to modify system behavior without changing code. They fall into various usage categories, and it's important to take that categorization into account when implementing and managing toggles. Toggles introduce complexity. We can keep that complexity in check by using smart toggle implementation practices and appropriate tools to manage our toggle configuration, but we should also aim to constrain the number of toggles in our system.

09 October 2017

Pete Hodgson

CONTENTS

expand

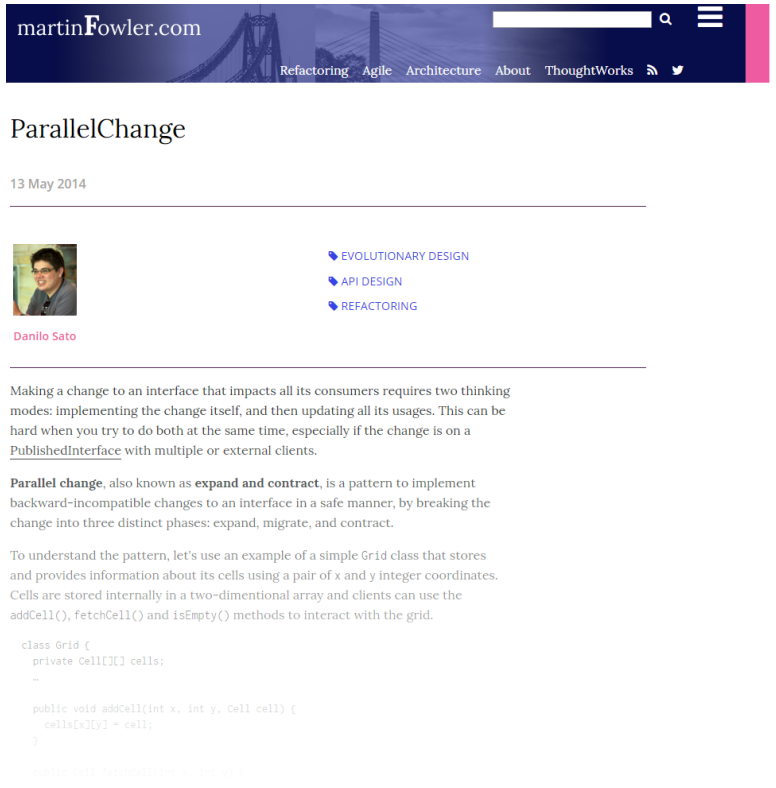
A Toggling Tale
Categories of toggles
Implementation Techniques
Toggle Configuration
Working with feature-flagged systems

Pete Hodgson is an independent software delivery consultant based in the San Francisco Bay Area. He specializes in helping startup engineering teams improve their engineering practices and technical architecture.

Pete previously spent six years as a consultant with ThoughtWorks, leading technical practices for their West Coast business. He also did several stints as a tech lead at various San Francisco startups.

POPULAR

martinfowler.com/articles/feature-toggles.html



martinFowler.com

Refactoring Agile Architecture About ThoughtWorks

ParallelChange

13 May 2014

Danilo Sato

EVOLUTIONARY DESIGN
API DESIGN
REFACTORING

Making a change to an interface that impacts all its consumers requires two thinking modes: implementing the change itself, and then updating all its usages. This can be hard when you try to do both at the same time, especially if the change is on a PublishedInterface with multiple or external clients.

Parallel change, also known as **expand and contract**, is a pattern to implement backward-incompatible changes to an interface in a safe manner, by breaking the change into three distinct phases: expand, migrate, and contract.

To understand the pattern, let's use an example of a simple `Grid` class that stores and provides information about its cells using a pair of `x` and `y` integer coordinates. Cells are stored internally in a two-dimentional array and clients can use the `addCell()`, `fetchCell()` and `isEmpty()` methods to interact with the grid.

```
class Grid {
  private Cell[][] cells;
  …

  public void addCell(int x, int y, Cell cell) {
    cells[x][y] = cell;
  }
```

martinfowler.com/bliki/ParallelChange.html

@_AlexYates_

#DevOps

Zero Downtime Database Deployments

DLM Consultants

Further reading...

@_AlexYates_

#DevOps

Zero Downtime Database Deployments

DLM Consultants

Alex Yates

Database DevOps Consultant

alex.yates@dlmconsultants.com
@_AlexYates_
workingwithdevs.com

MVP Microsoft® Most Valuable Professional

DATA RELAY

FRIEND OF redgate 2019

@_AlexYates_
#DevOps

Zero Downtime Database Deployments

DLM Consultants