



# Optimizing Canary Deployments for user-centric success

**Bob Walker | Field CTO | Octopus Deploy**

@bobjwalker

[bob.walker@octopus.com](mailto:bob.walker@octopus.com)

@bobjwalker

bobjwalker.octopus.app

# Bob Walker



Field CTO  
Octonaut since 2018

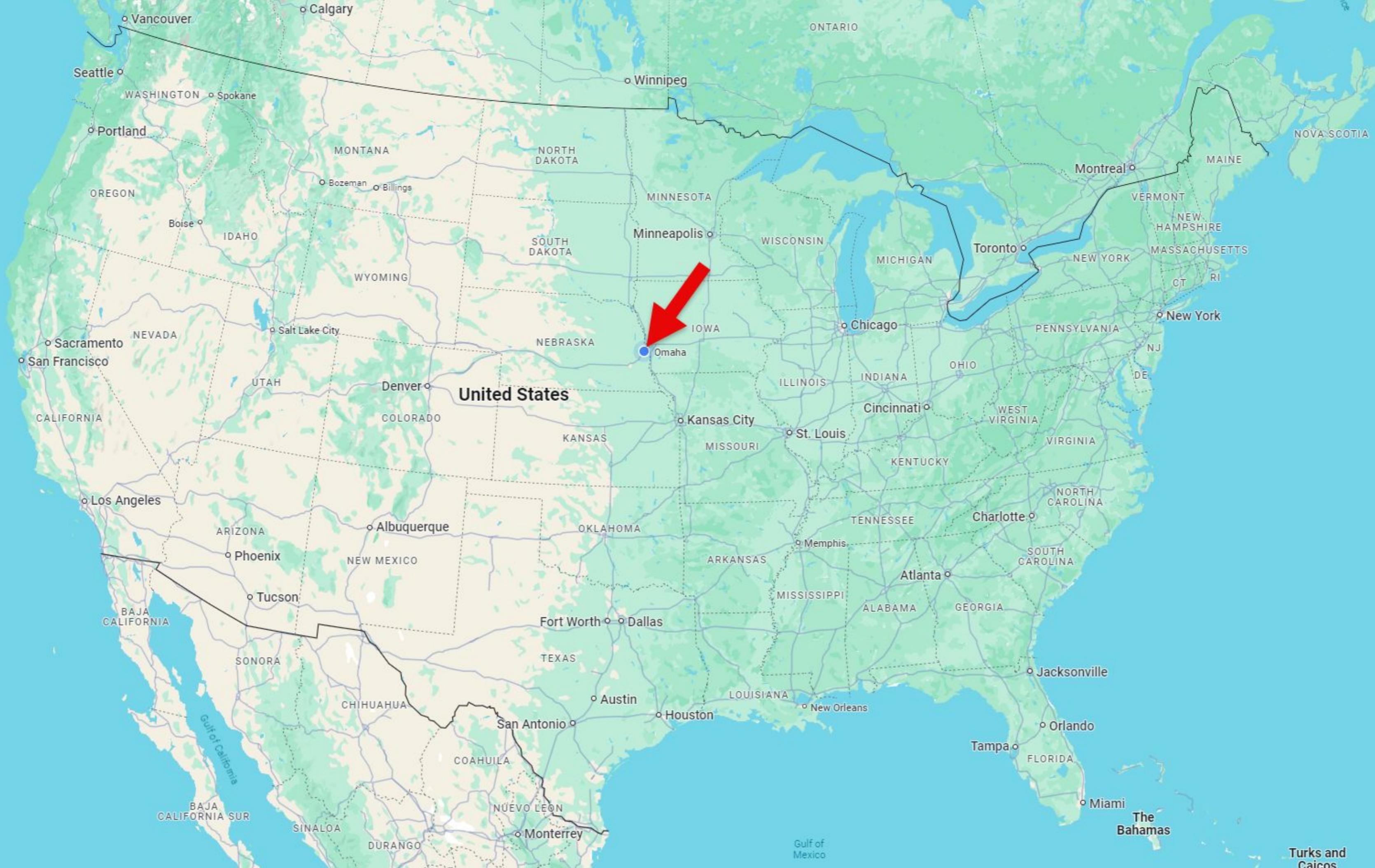
 @bobjwalker

 [bob.walker@octopus.com](mailto:bob.walker@octopus.com)

 @bobjwalker

 [bobjwalker.octopus.app](http://bobjwalker.octopus.app)







Farm Credit Services *of America*

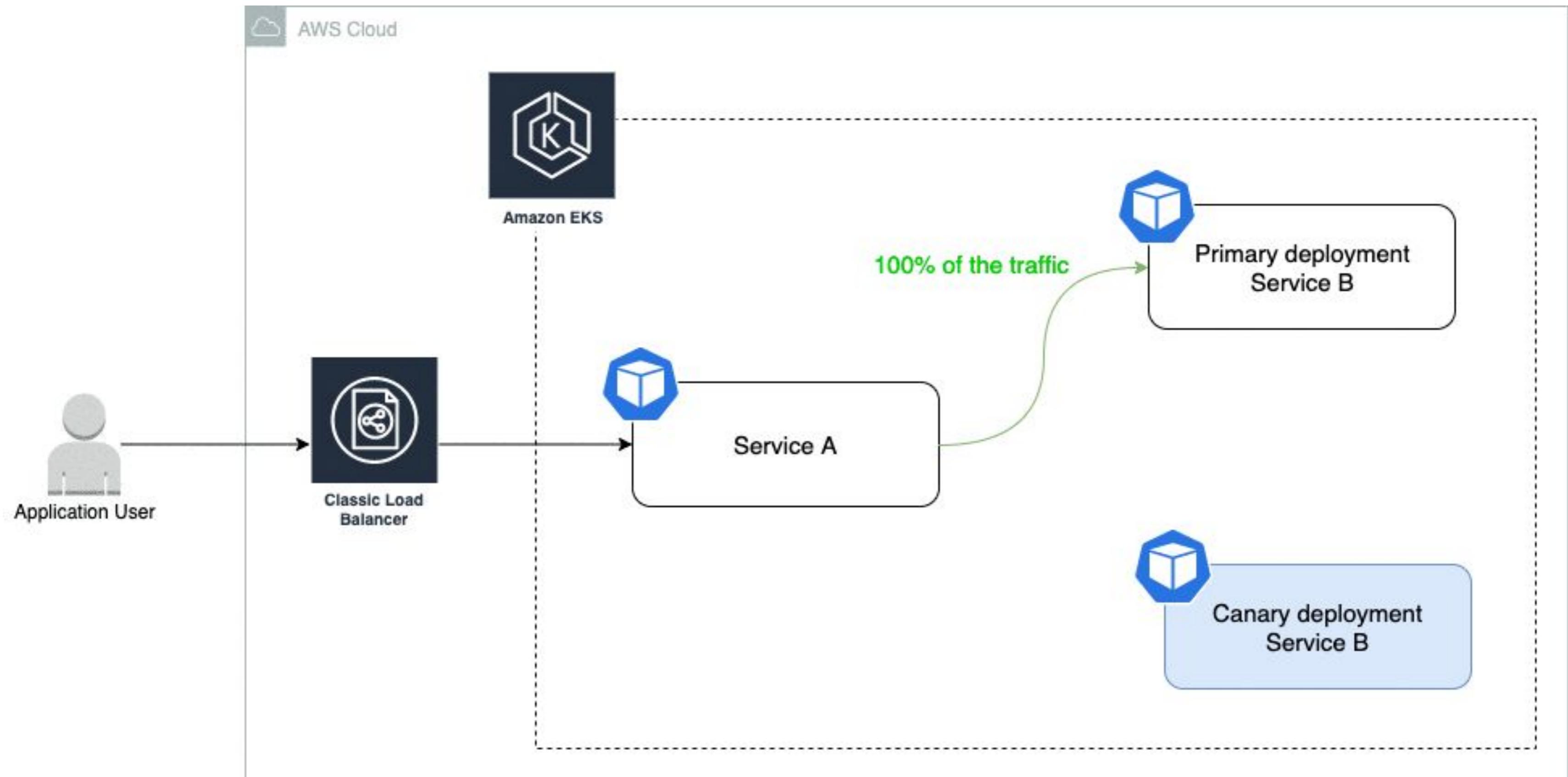


Octopus Deploy



**Serverless, PaaS, Containers and  
Kubernetes are helping to drive  
progressive delivery solutions**

You are likely thinking about  
canary deployments wrong



You have to consider user and business requirements along with application type, architecture, and host.

**Business Requirements are NOT  
Deployment Strategies**

Today we will discuss three  
canary deployment strategies

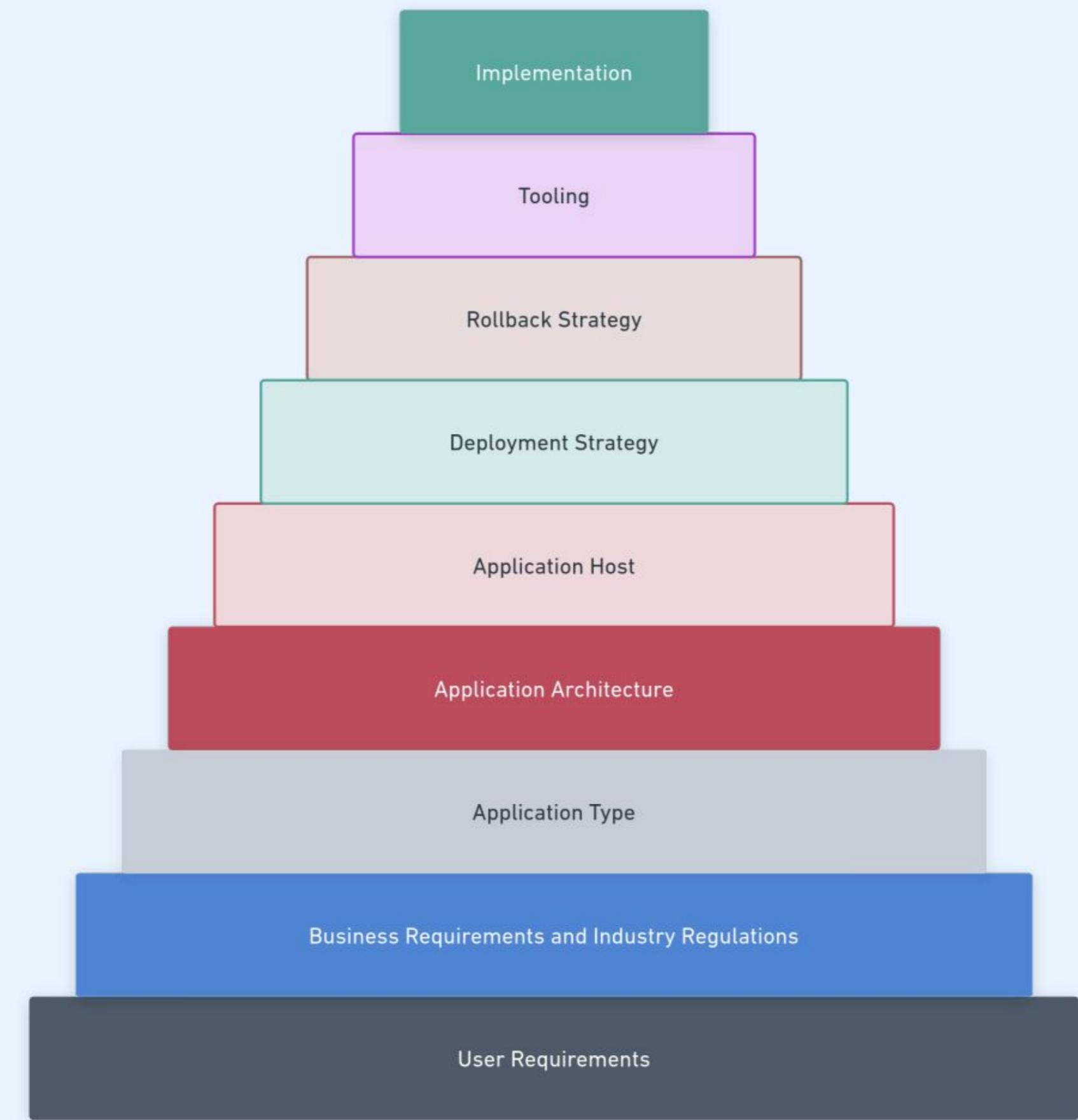
**Canary deployments are not for  
everyone - they require a lot  
changes to your applications and  
SDLC**

# Agenda

- Business Requirements
- Challenges of traffic based canary deployments
- A user centric approach
- Process changes

# Business Requirements





*The hierarchy of considerations.*

# Business Requirements

- Uptime
  - US-based online bank must be available 24 hours daily, Monday to Saturday
  - Grocery store must be online during opening hours
  - Octopus Cloud must respect customer maintenance windows
- Reduce risk of new bugs
  - The code doesn't have show-stopping bugs
  - Fix bugs and issues before code reaches wider audience
  - Make it easy to rollback
- Ensuring new business functionality works
  - Test and gather feedback before releasing to public
  - Limit the impact of any potential changes

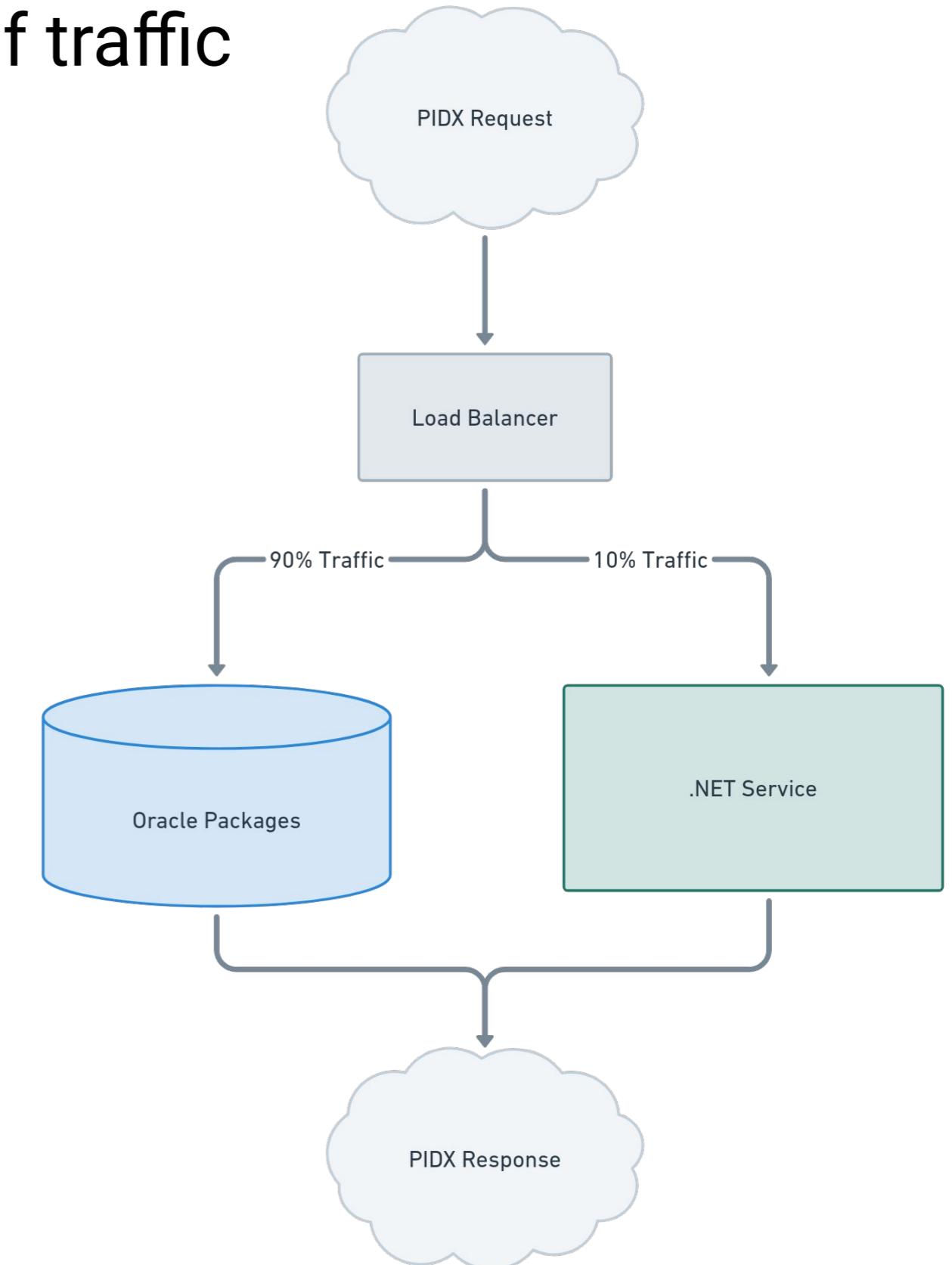
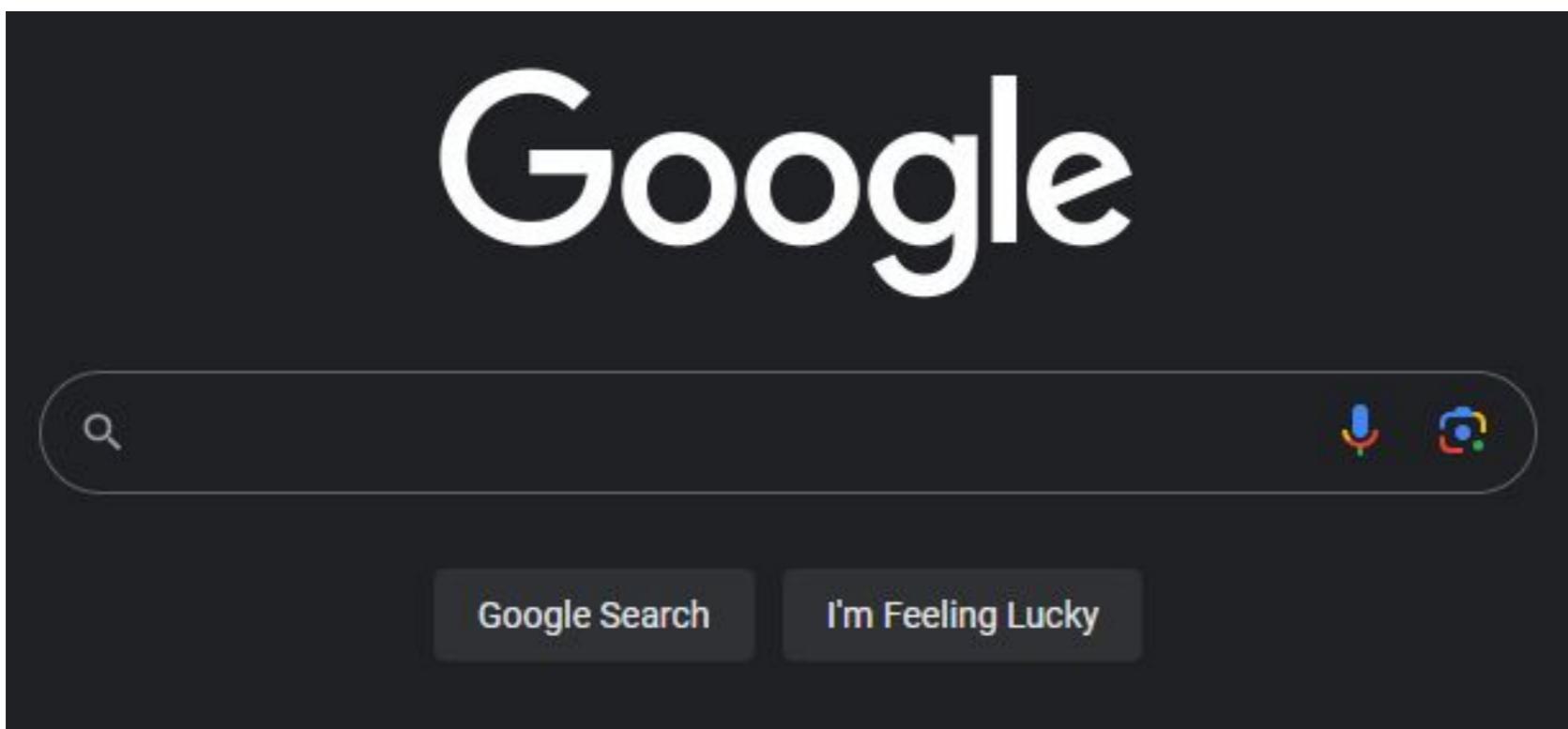
# Challenges of Traffic-Based Canary Deployments



Traffic-based canary deployments  
is appropriate for a subset of use  
cases.

# Appropriate Use Cases

- The application processes a significant amount of traffic
- Transactional - sessions do not matter
- Moving / upgrading hosts



**Indiscriminately routing a  
percentage of traffic for new  
functionality leads can lead to a  
disjointed user experience**

# Inappropriate Use Cases

- Changing of business rules
- Interactive applications
- Updating the user interface
- A/B Experiments





Target - Kubern...



Dashboard

Projects



Infrastructure    Tenants    Insights    Library    Tasks

Dashboard



Default



Type Ctrl+Space to search or switch projects...

Filter by project group



All Projects



Tenants



Tenant Tag Sets



Variable Sets



OVERVIEW



Deployment Targets



Environments



Machine Policies



Machine Proxies



Workers



Worker Pools



MANAGE



Accounts



Build Information



Certificates



External Feeds



Git Credentials



Lifecycles



Packages



Script Modules



Step Templates

All Projects

Search projects...

Advanced filters

Filters applied

Default Project Group

Development

Test

Staging

Production

Pet Shop 2024.3.1.1-ClinicsBrand 2024.3.0.9 1/1 Mar 11, 2024 3:22 PM 1/1

2024.3.0.9 Mar 11, 2024 3:41 PM 4/4

2024.3.0.9 Mar 11, 2024 3:35 PM 5/12

Random Quotes 0.1.42 Feb 27, 2024 3:01 PM 0.1.79.1 Feb 27, 2024 3:24 PM

0.1.79.1 Feb 27, 2024 3:26 PM

0.1.79.1 Feb 27, 2024 3:30 PM

Trident 4.0.56-helm.9 Apr 17, 2024 9:55 AM 4.0.56.466 Apr 17, 2024 10:21 AM

4.0.56.466 Apr 17, 2024 10:27 AM

4.0.43.421 Mar 19, 2024 8:56 AM

**Session affinity or sticky sessions  
can help  
(However...)**

# Sticky sessions won't solve

- Users who leverage multiple devices
- Rollouts that last longer than the session duration
- Applications used by groups of users
  - Subset of users sessions see new functionality while others do not
  - B2B SaaS
  - Deploying to edge nodes (factories, retail locations, branch offices, banks)
- Business rule coupling
  - The services and applications are independently deployable
  - Business rule change requires updating multiple services / applications
  - They each implement a different portion of the business rule

# What are our requirements?

# Business Requirements

- Uptime
  - US-based online bank must be available 24 hours daily, Monday to Saturday
  - Grocery store must be online during opening hours
  - Octopus Cloud must respect customer maintenance windows
- Reduce risk of new bugs
  - The code doesn't have show-stopping bugs
  - Fix bugs and issues before code reaches wider audience
  - Make it easy to rollback
- Ensuring new business functionality works
  - Test and gather feedback before releasing to public
  - Limit the impact of any potential changes

# A user centric approach

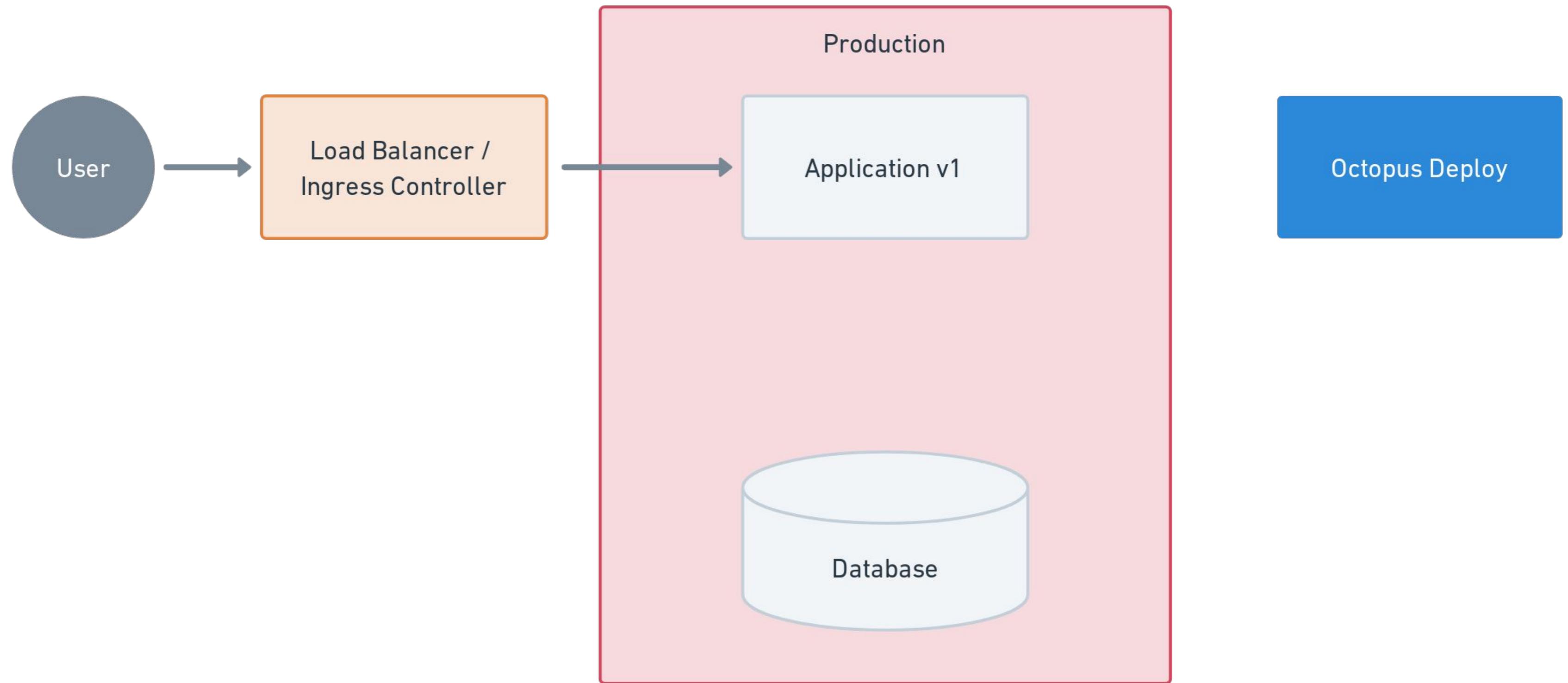


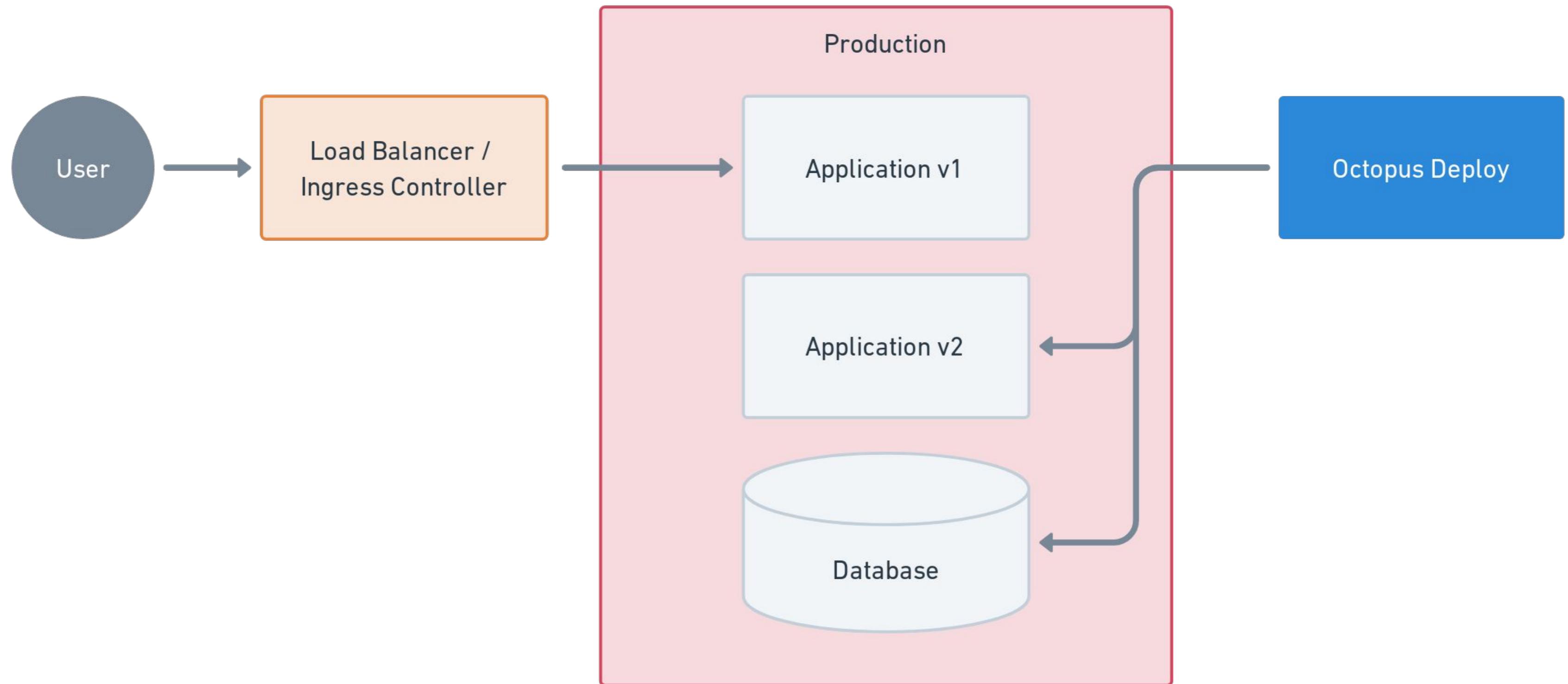
# Three Canary Deployment Options

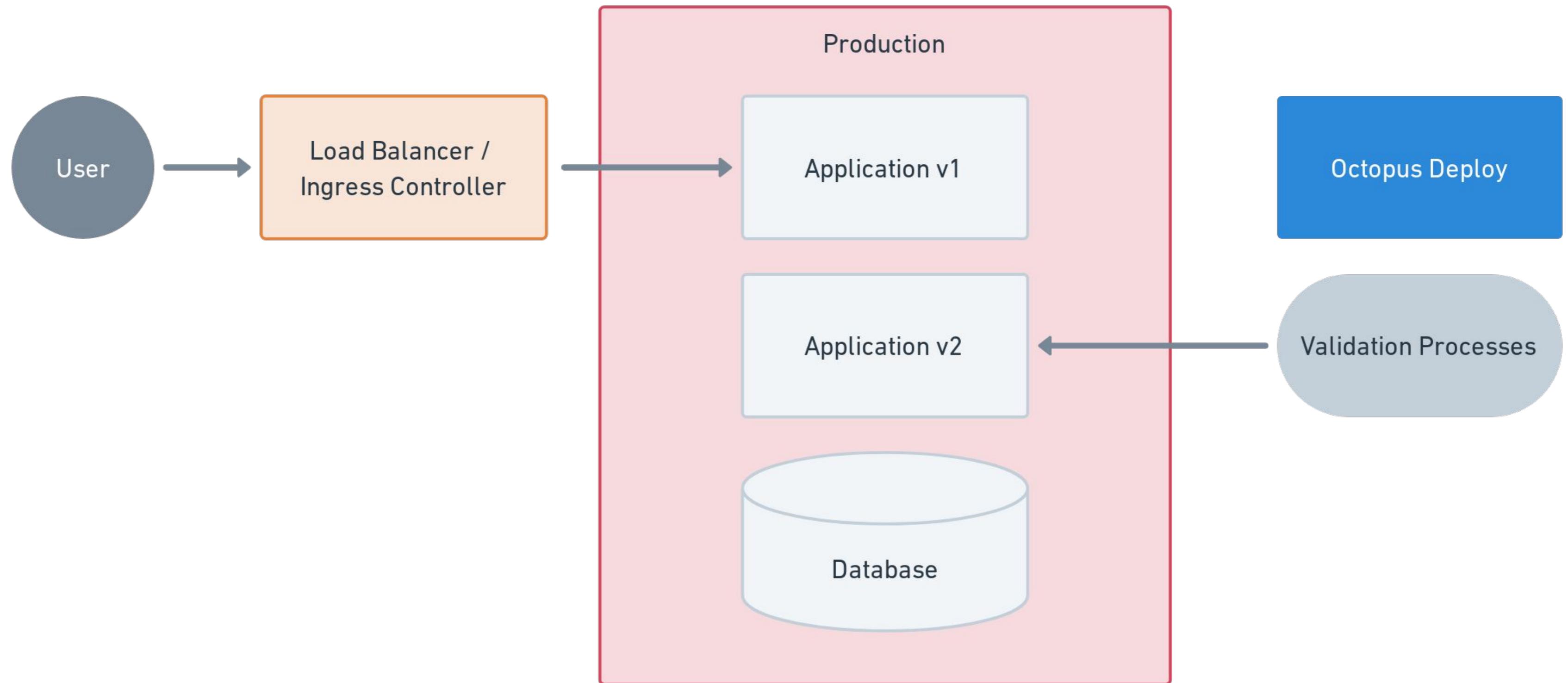
- Traffic-based routing
- Feature flags (or release toggles)
- Release rings

# **Feature Flags**

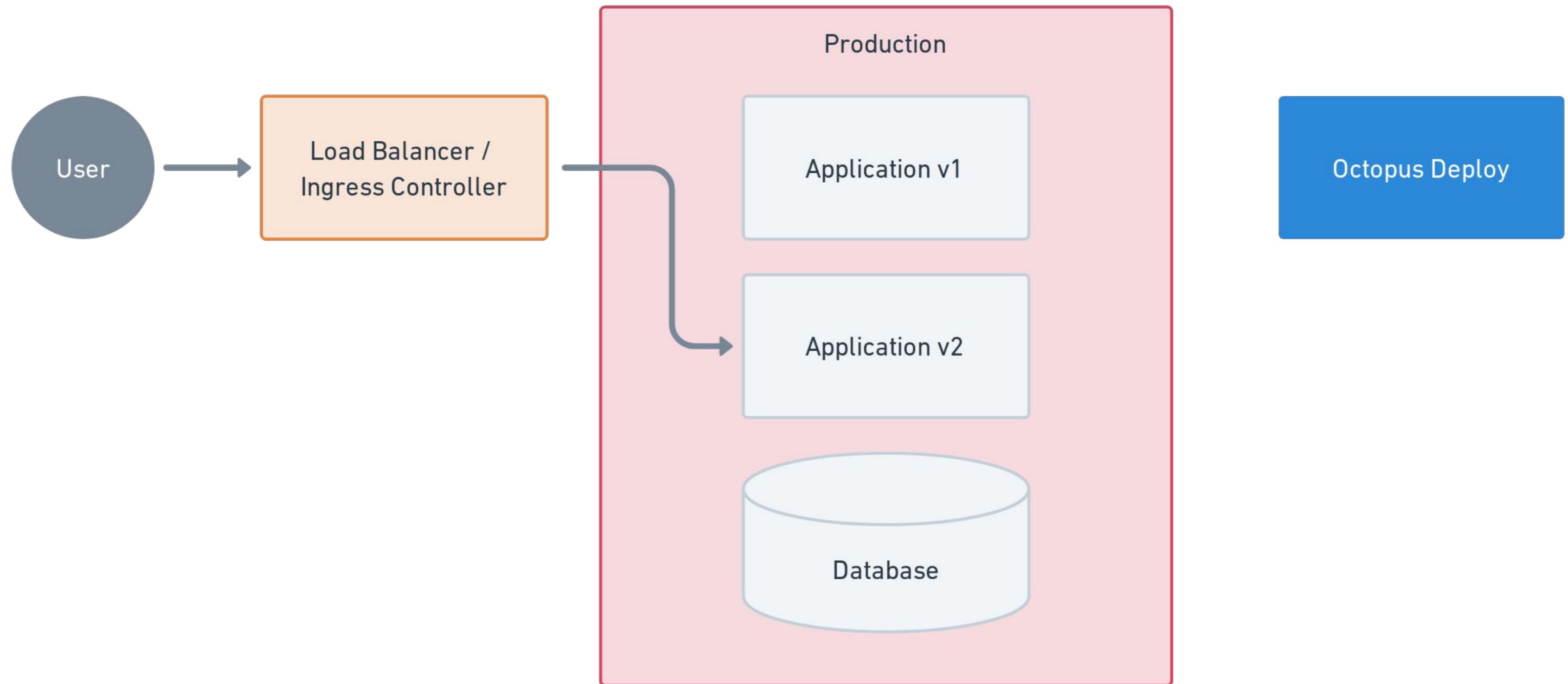
**(combined with a Blue/Green or Staging Pattern)**







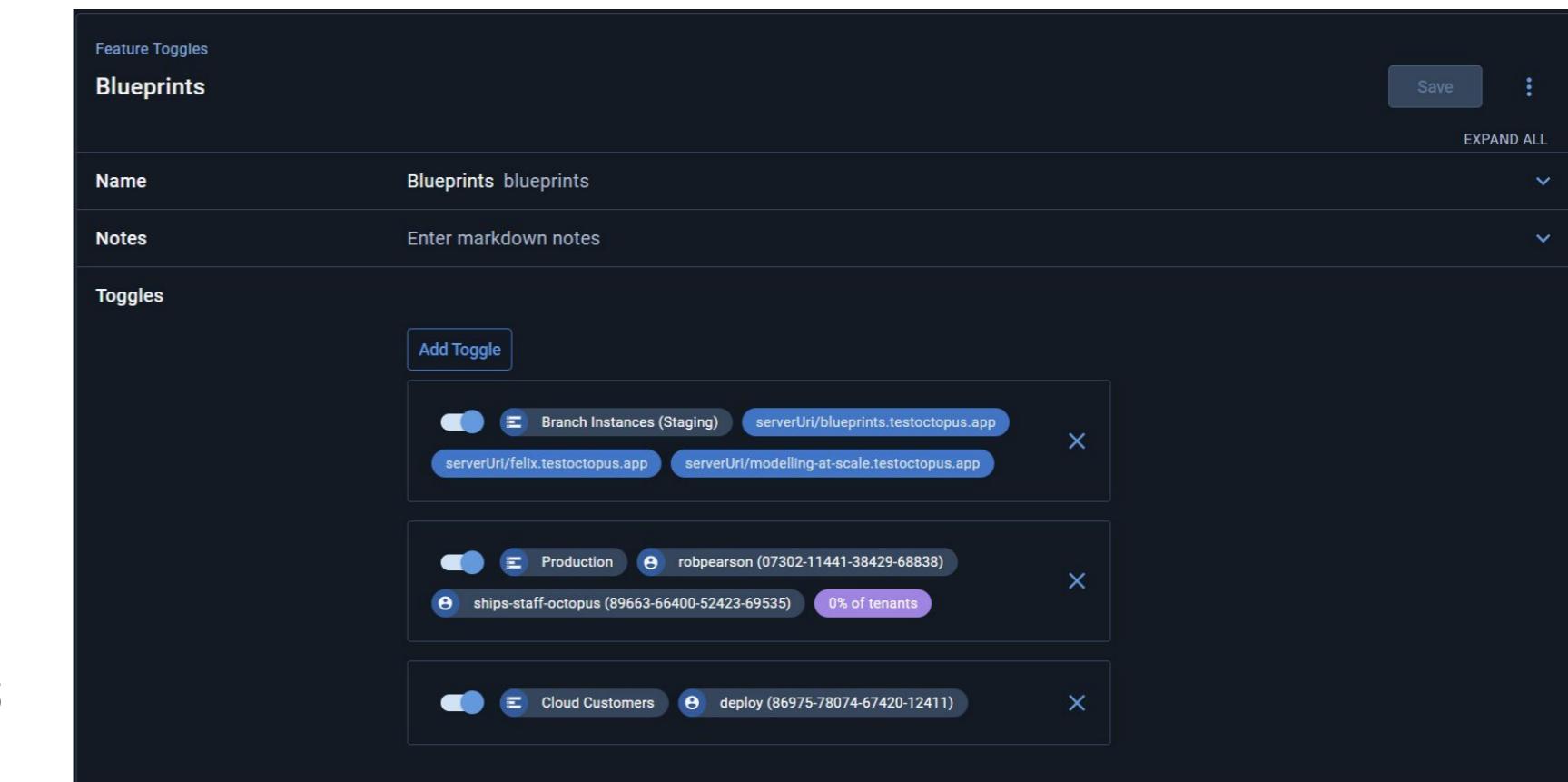
If a problem is discovered during validation don't update the load balancer



**Separate deploying the code and  
releasing new features**

# Feature Flags

- Segments - Deploy with flag turned off
- Internal - Enable all developers for testing / verification
- Pick a subset of users
  - B2C applications - users can “opt-in”
  - Internal applications - target specific teams verification
  - B2B applications - target specific companies
- General Adoption
  - Expand to more users / groups / companies
  - Enable for everyone over period of time



**If a problem is discovered,  
stop enabling the feature flags  
(if required - turn them off)**

# Helps Prevent “Big-Bang” merges

When you are gonna merge two branches after long time



# Release rings

# Release Rings

- Release rings - Alpha, Beta, General Adoption
- B2B SaaS and/or Isolated Infrastructure
  - Alpha - Deploy to developers and internal users
  - Beta - Pick a subset of customers - either via opt-in or randomized list
  - General Adoption - Rollout to more companies and then all companies

Deployments				
Release Ring:	Development	Test	Staging	Production
Edge				
Internal	<input checked="" type="checkbox"/> 2024.3.1.1-ClinicsBranding Mar 11, 2024 3:22 PM	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 11, 2024 3:26 PM	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 11, 2024 3:31 PM	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 11, 2024 3:35 PM
Early Adopters				
NE01-Omaha	🚫	🚫	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 11, 2024 3:41 PM	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 19, 2024 2:53 AM
UT01-Salt Lake City	🚫	🚫	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 11, 2024 3:41 PM	Filter required
UT02-Logan	🚫	🚫	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 11, 2024 3:41 PM	Filter required
Stable				
NE02-Lincoln	🚫	🚫	🚫	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 19, 2024 2:53 AM
NE03-Benson	🚫	🚫	🚫	<input checked="" type="checkbox"/> 2024.3.0.9 Mar 22, 2024 6:23 AM
NE04-Grand Island	🚫	🚫	🚫	Filter required
NE05-Niobrara	🚫	🚫	🚫	Filter required

**If a problem is discovered,  
stop rolling out to other rings**

# Use Cases

# Internal Experience - Octopus Cloud

- Isolated infrastructure for each customer
- Update instances via release rings and leverage feature flags
  - Alpha - Deploy to developers and internal users
  - Beta - Randomly select a subset of customers
  - General Adoption - Rollout to remaining customers
  - Feature flags limit new functionality to subset of customers
- Update all ~2000 instances in 48 hours
  - Andon Cord - stops updating instances
  - Feature flags disabled for weeks for most customers
  - Feature flags are removed before self-hosted release

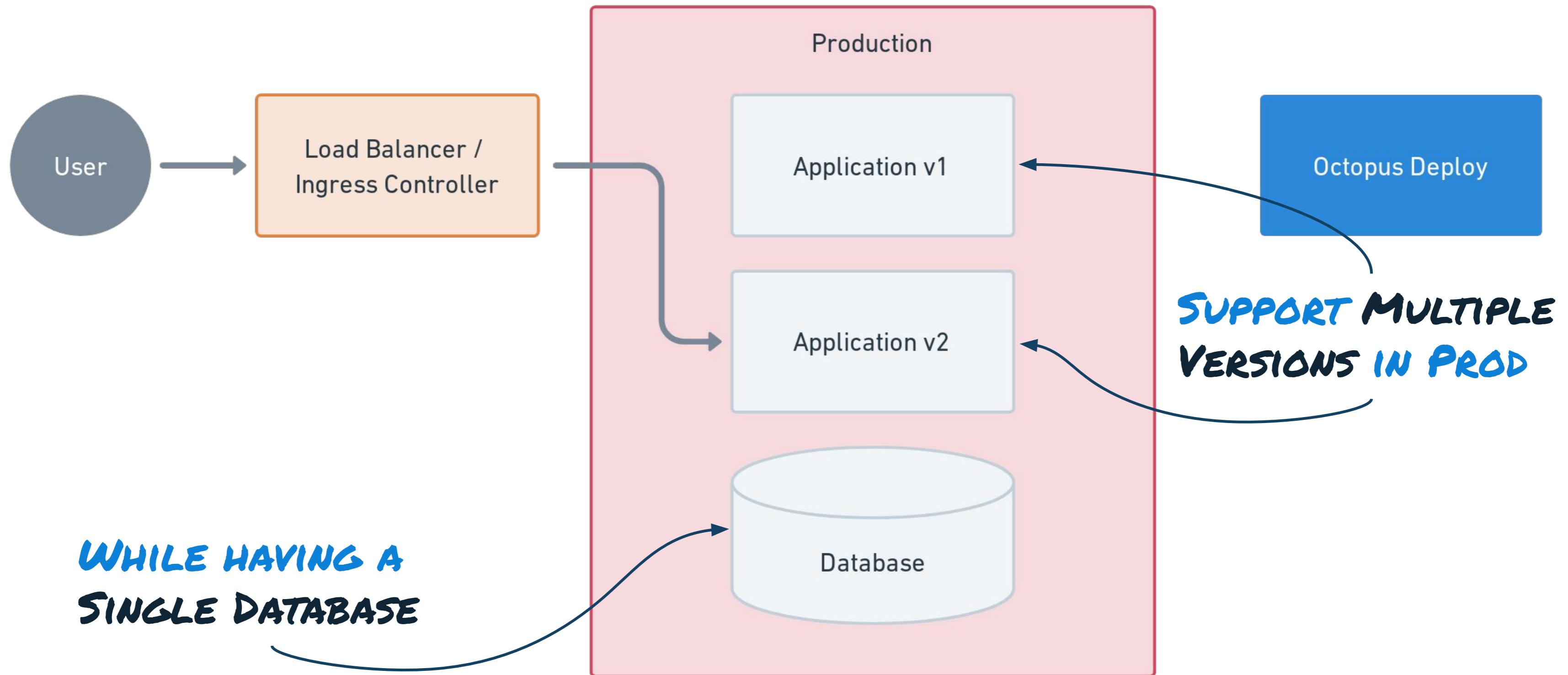
# Deployments to hospitals

- Each hospital runs a dedicated copy of the application
- Update instances via release rings
  - Alpha - Deploy to a subset of locations
  - Beta - Increase the number of locations
  - General Adoption - Rollout to the remaining locations
- Challenges
  - Cannot take everything offline
  - Need to wait a while to let the new functionality “bake-in”

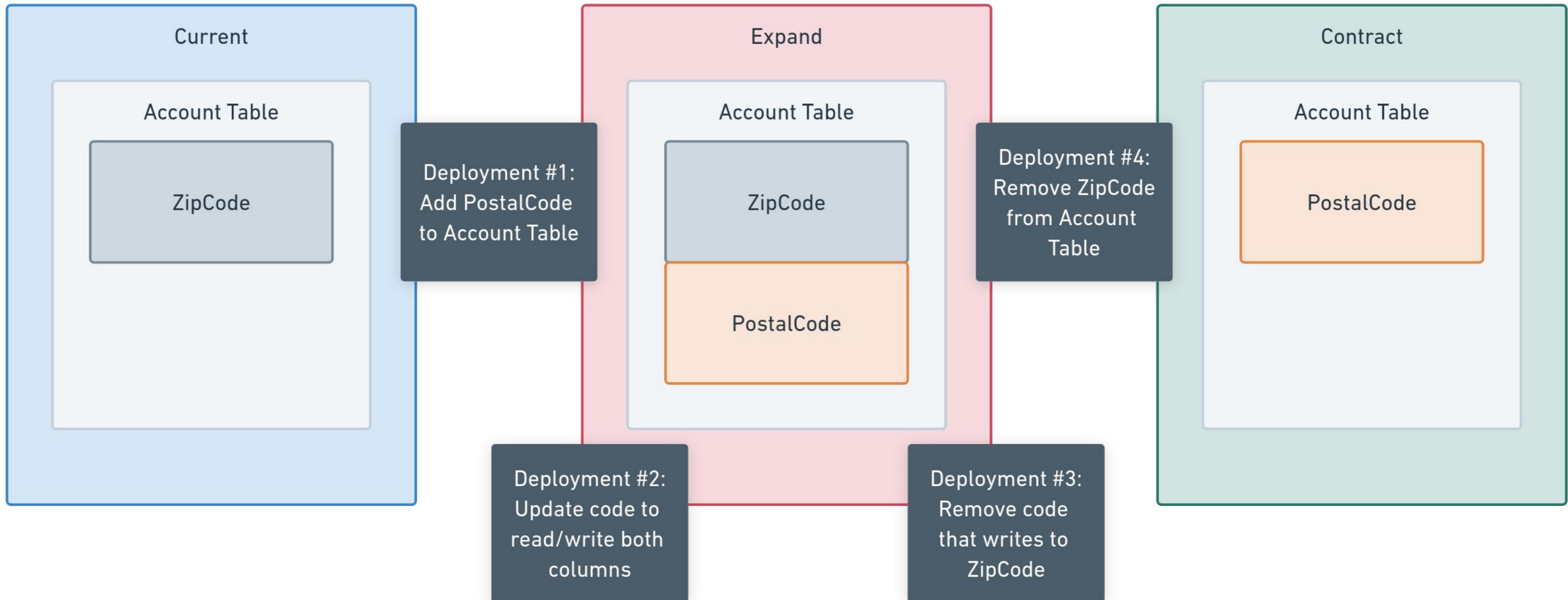
# Process Changes



# Changes your SDLC



# Expand and Contract Pattern



# Feature Flags / Toggles

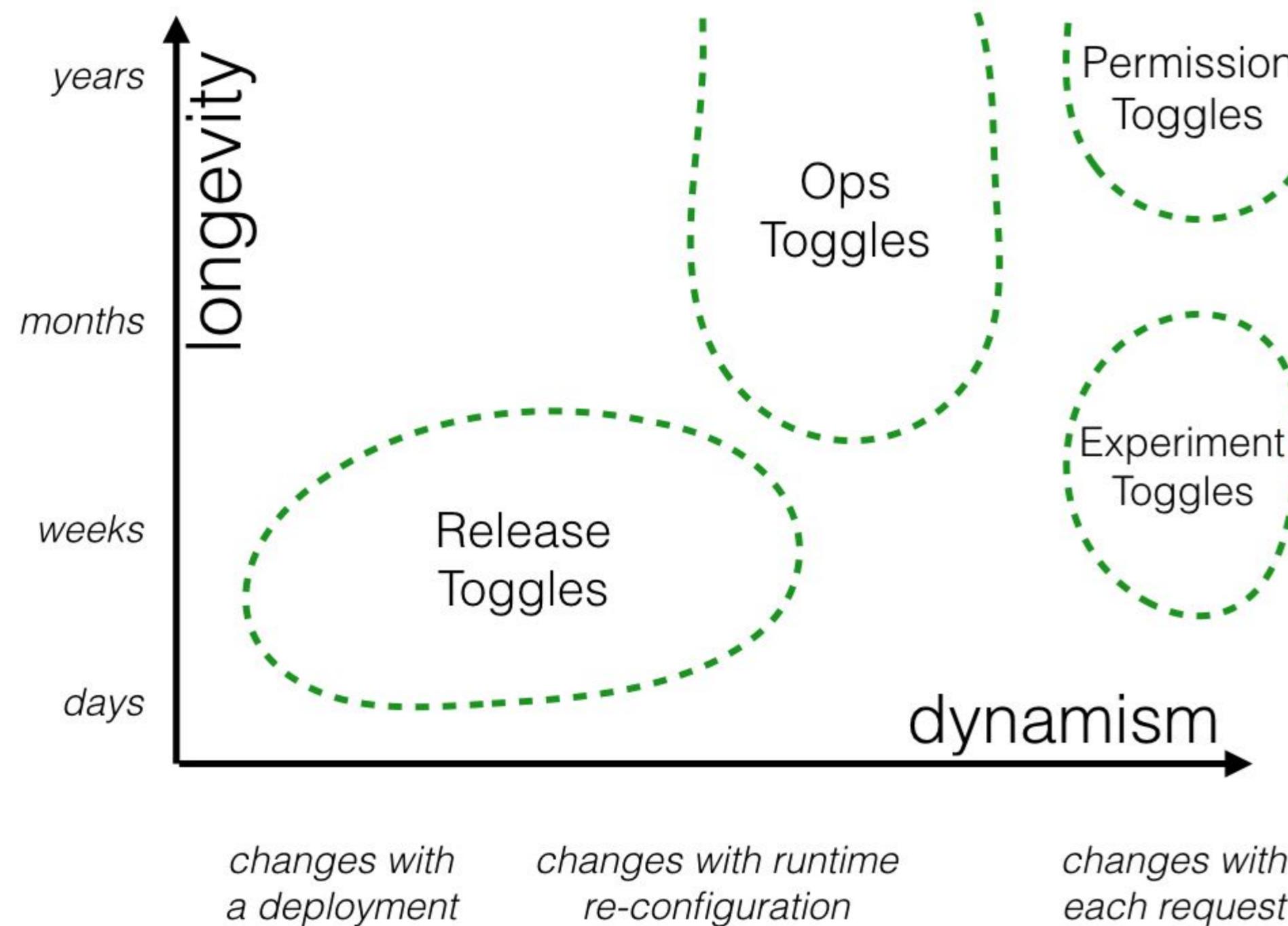


Image Source: <https://martinfowler.com/articles/feature-toggles.html>

## OpenFeature

Standardizing Feature Flagging for Everyone

[Learn more](#)

[Get started](#)

```
OpenFeature.setProvider(new MyProvider());  
  
const featureFlags = OpenFeature.getClient();  
  
const withCows = await featureFlags.getBooleanValue("with-cows", false);  
if (withCows) {  
  res.send(cowsay.say({ text: "Hello, world!" }));  
} else {  
  res.send("Hello, world!");  
}
```

ultimate flexibility

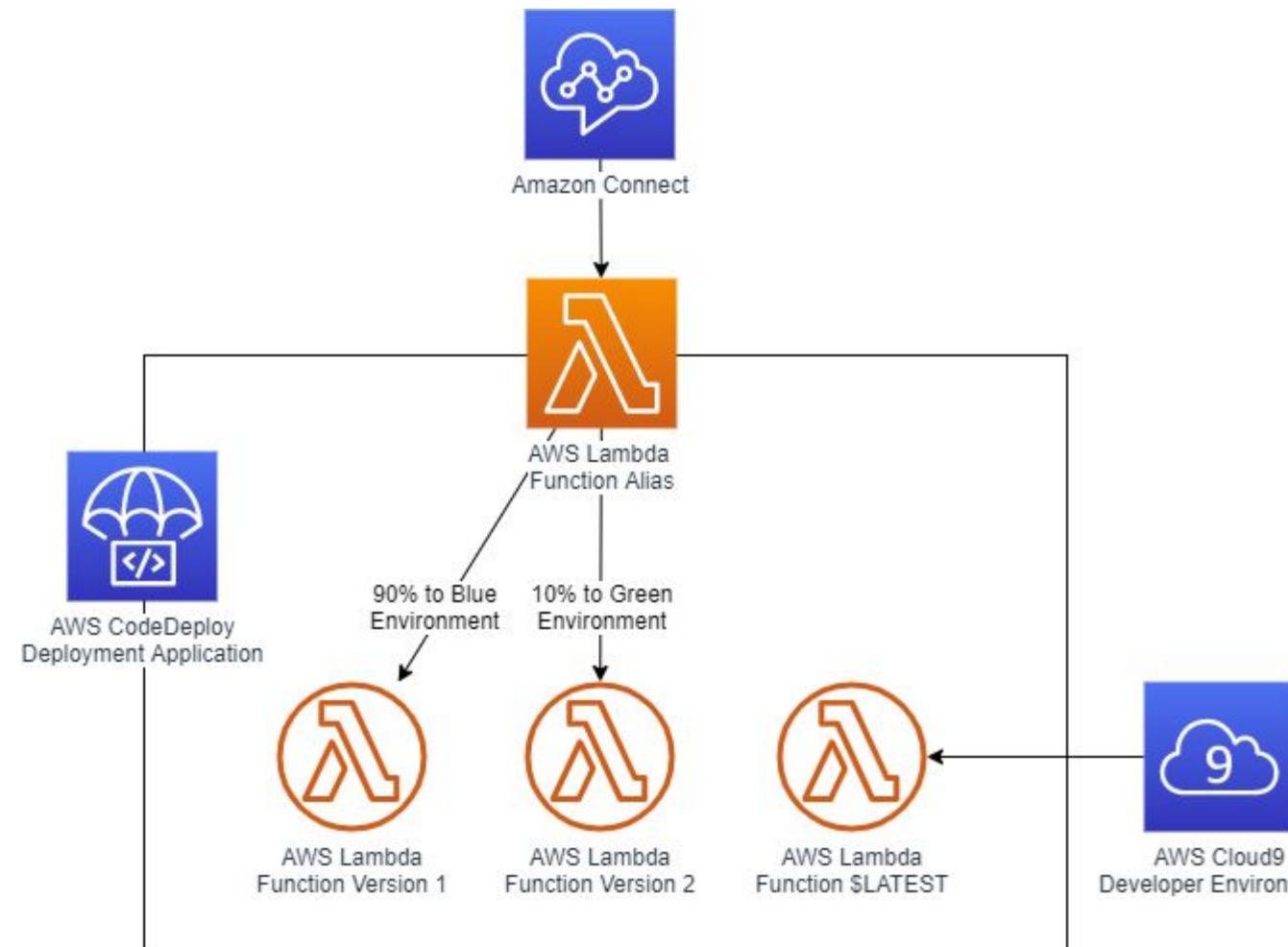
**Speaks your language**

OpenFeature supports many of the top programming languages, with more on the way.

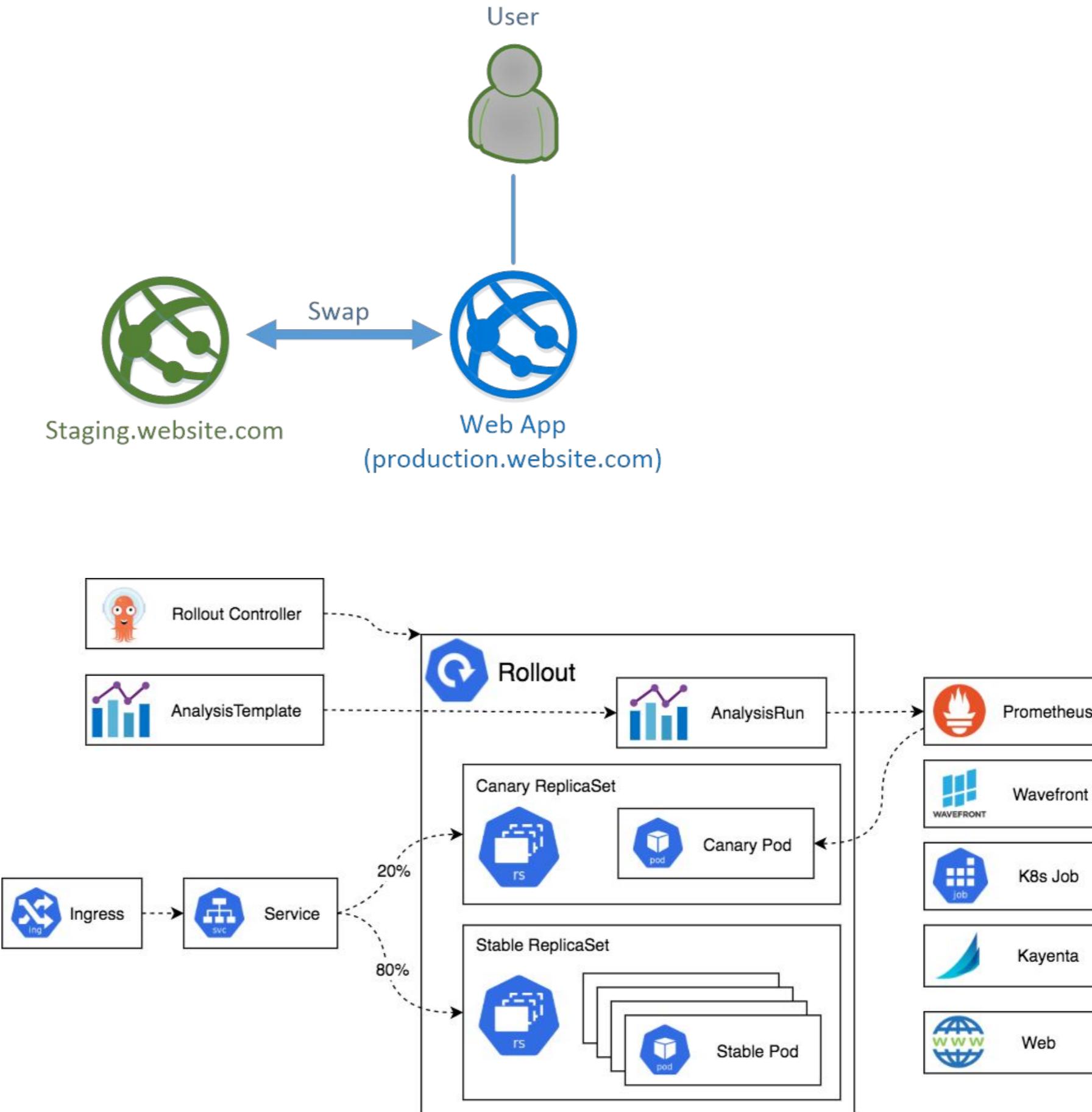


<https://openfeature.dev>

# Deployments depend on your application platform



**DON'T GO AGAINST  
THE GRAIN!!!!**



**Canary deployments will require a significant investment**

**Focus on user and business requirements along with application type, architecture, and host.**

Only Michael Jordan can be  
Michael Jordan - you be you

*Caitlin Clark*

Only ~~Michael Jordan~~ can be  
~~Michael Jordan~~ - you be you

*Caitlin Clark*

# (Free) White Paper for Deployment Strategies



<https://octopus.com/whitepapers>



## Modern deployment and rollback strategies

Bob Walker



# Thank you!

**Bob Walker - Field CTO at Octopus Deploy**

@bobjwalker

[bob.walker@octopus.com](mailto:bob.walker@octopus.com)

@bobjwalker

bobjwalker.octopus.app

