
AWS
re:Invent

DOP311

Share the road with traffic splitting and shared load balancers

Christian Weber

Developer Advocate

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Introductions

New to AWS Elastic Beanstalk?

Traffic splitting

Shared load balancing



AWS re:Invent

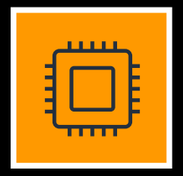
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

New to Elastic Beanstalk?



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS offers many options



Compute: Amazon EC2, AWS Lambda, Amazon ECS, Amazon EKS, Amazon Lightsail, AWS Fargate



Storage: Amazon S3, Amazon Elastic File System



Network: Classic Load Balancer, Application Load Balancer, Network Load Balancer, Security Groups, Amazon Virtual Private Cloud

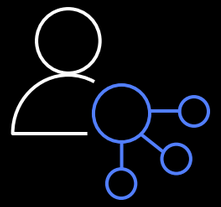


Database: Amazon RDS, Amazon DynamoDB, Amazon Aurora, Amazon Neptune



Management: AWS Auto Scaling, AWS CloudFormation, AWS Systems Manager, Amazon CloudWatch

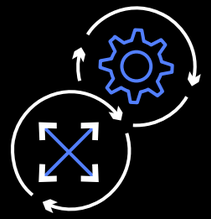
Easy to get started



Upload

Multiple ways to upload source bundle

- EB console
- EB CLI
- AWS Extensions for .NET CLI
- AWS Toolkit for Visual Studio
- Windows Web App Migration Assistant (.NET apps)
- CI / CD integration



Configure

Select from 11 runtime platforms, including

- Python
- Java
- Node.js
- .NET
- Docker
- PHP
- Apache Tomcat

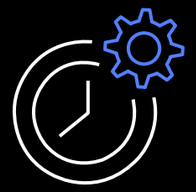


Deploy

One-click deployment automatically provisions

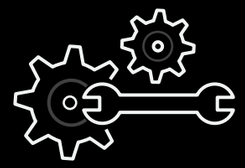
- Auto scaling
- Load balancing
- Network configuration
- App deployment

Gateway to modernization



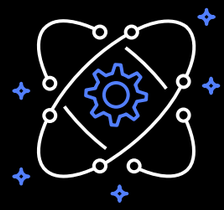
Automate deployments with CI / CD

- Automatically deploy to Elastic Beanstalk with each code commit
- Create CI / CD pipeline using AWS CodePipeline / Elastic Beanstalk integration
- Integrate Elastic Beanstalk into existing CI / CD pipeline



Replatform from Windows to Linux

- Run .NET Core apps in Elastic Beanstalk Linux environments and save 30% or more vs. Windows
- Convert apps from .NET Framework to .NET Core to run on Elastic Beanstalk / Linux



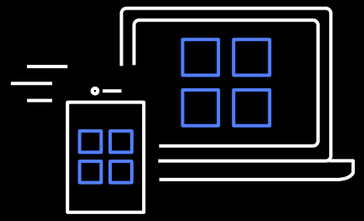
Modernize with containerization

- Containerize app automatically
- Gradually adopt service-oriented architecture

When do I use Elastic Beanstalk?



API services



Web apps and websites



Worker nodes

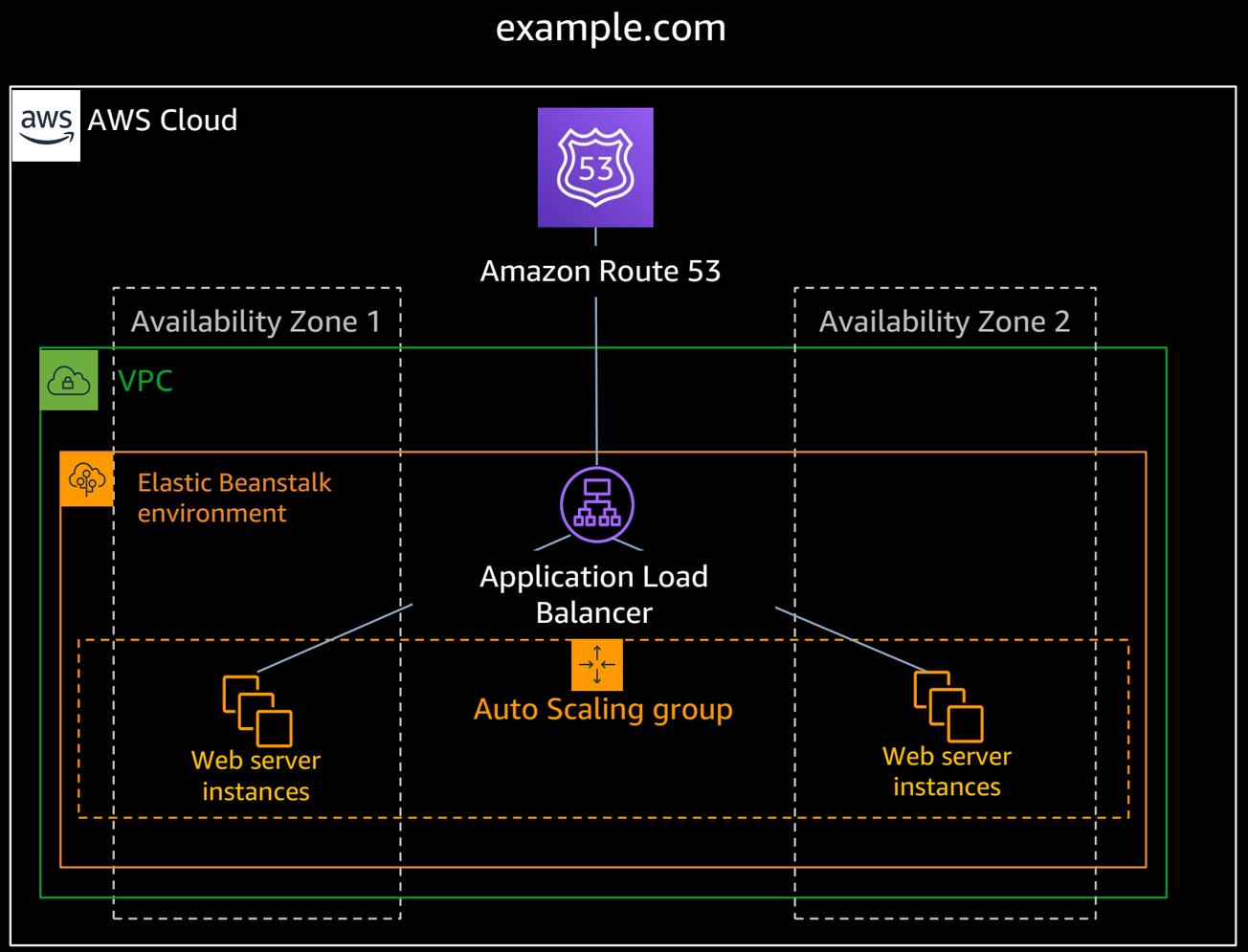
Elastic Beanstalk traffic splitting

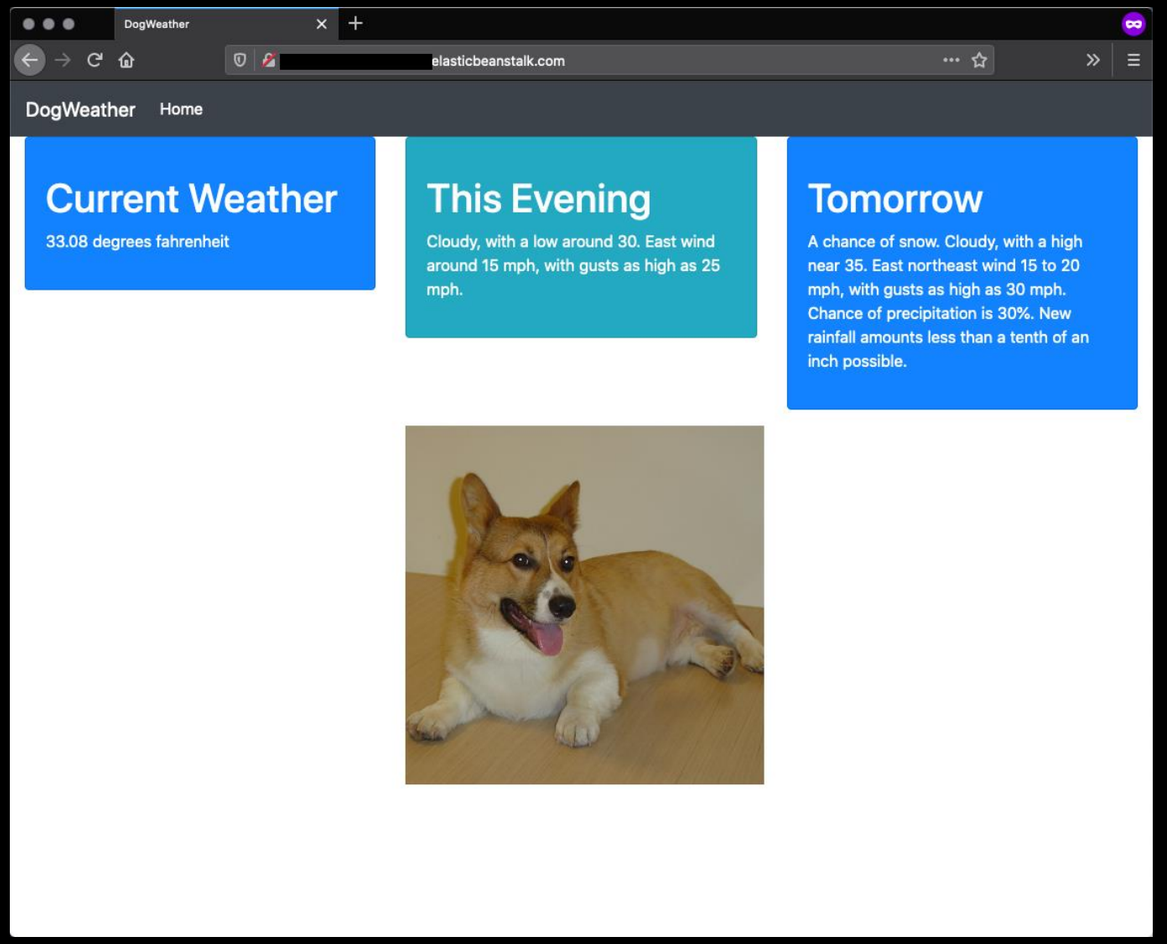


AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DogWeather Architecture





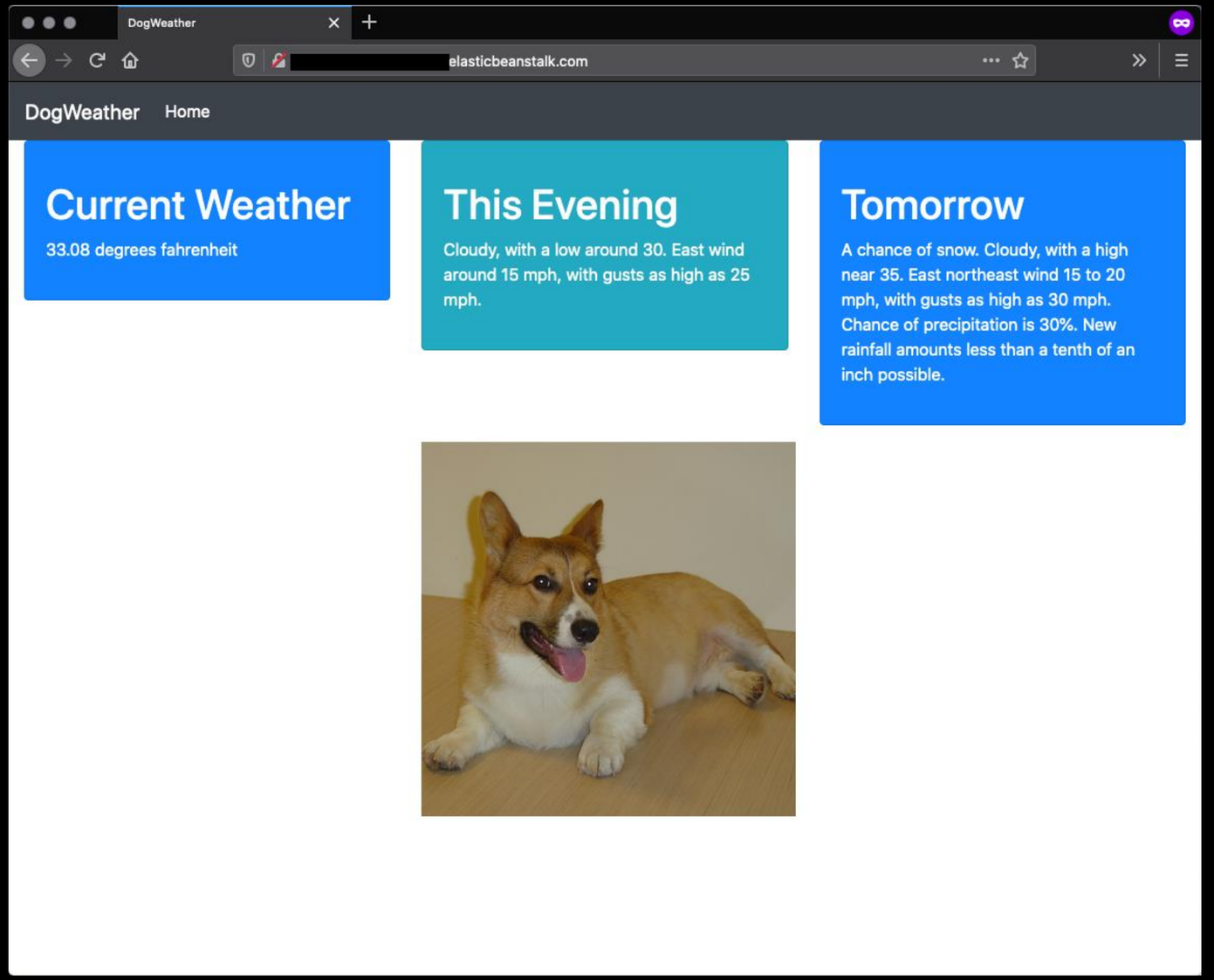
Scenario: Introducing a change to production



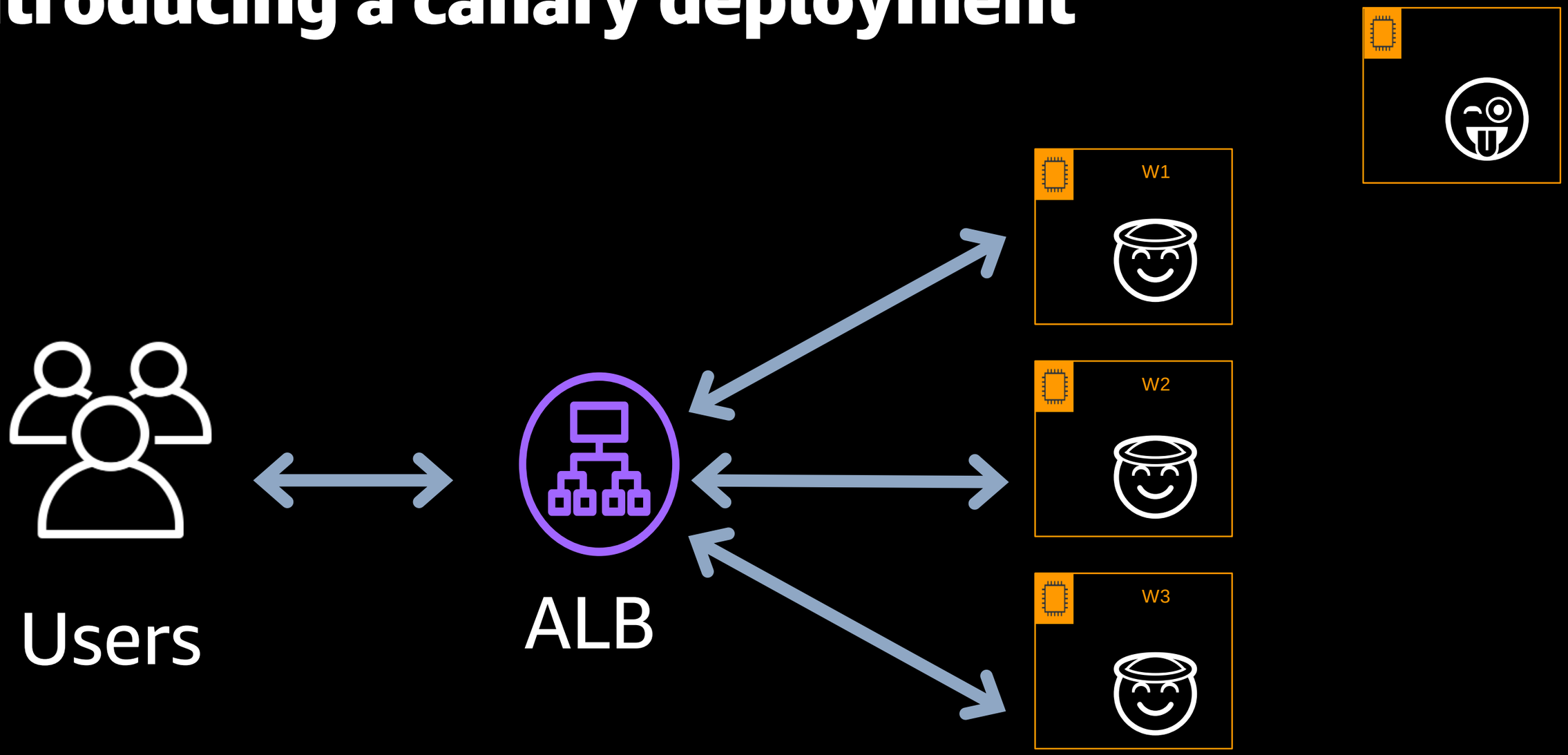
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

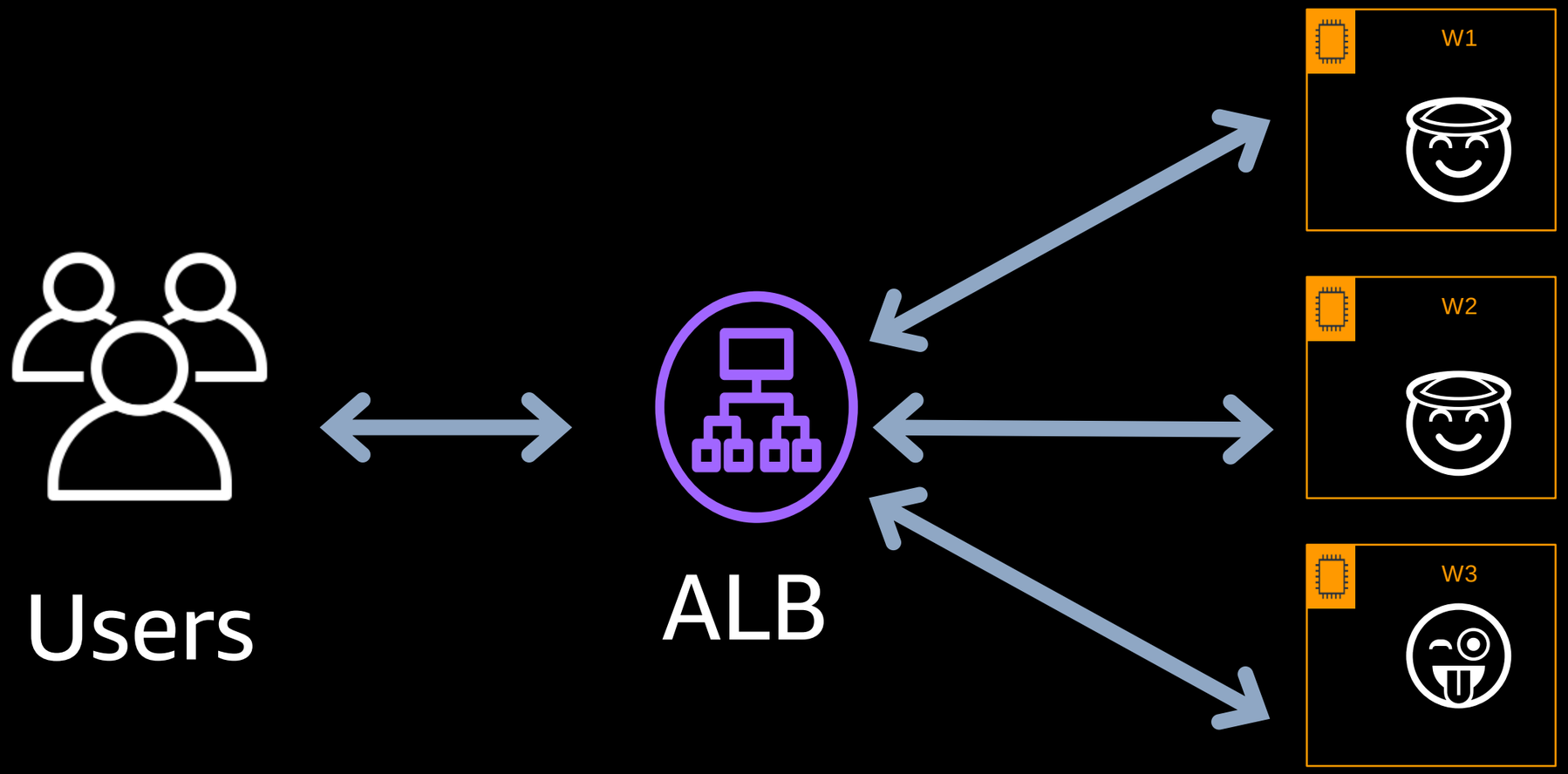
Helping friends see our site better



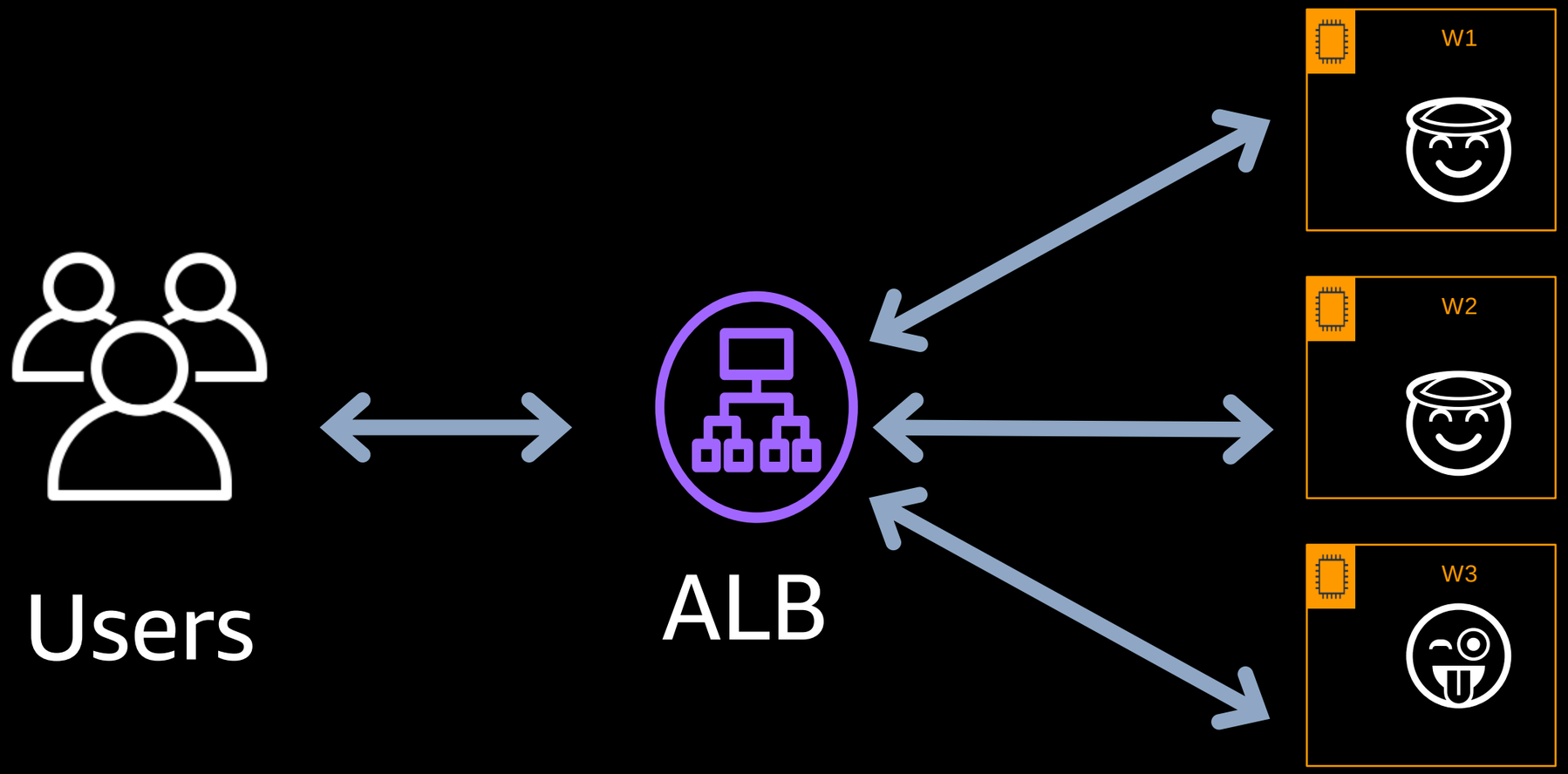
Introducing a canary deployment



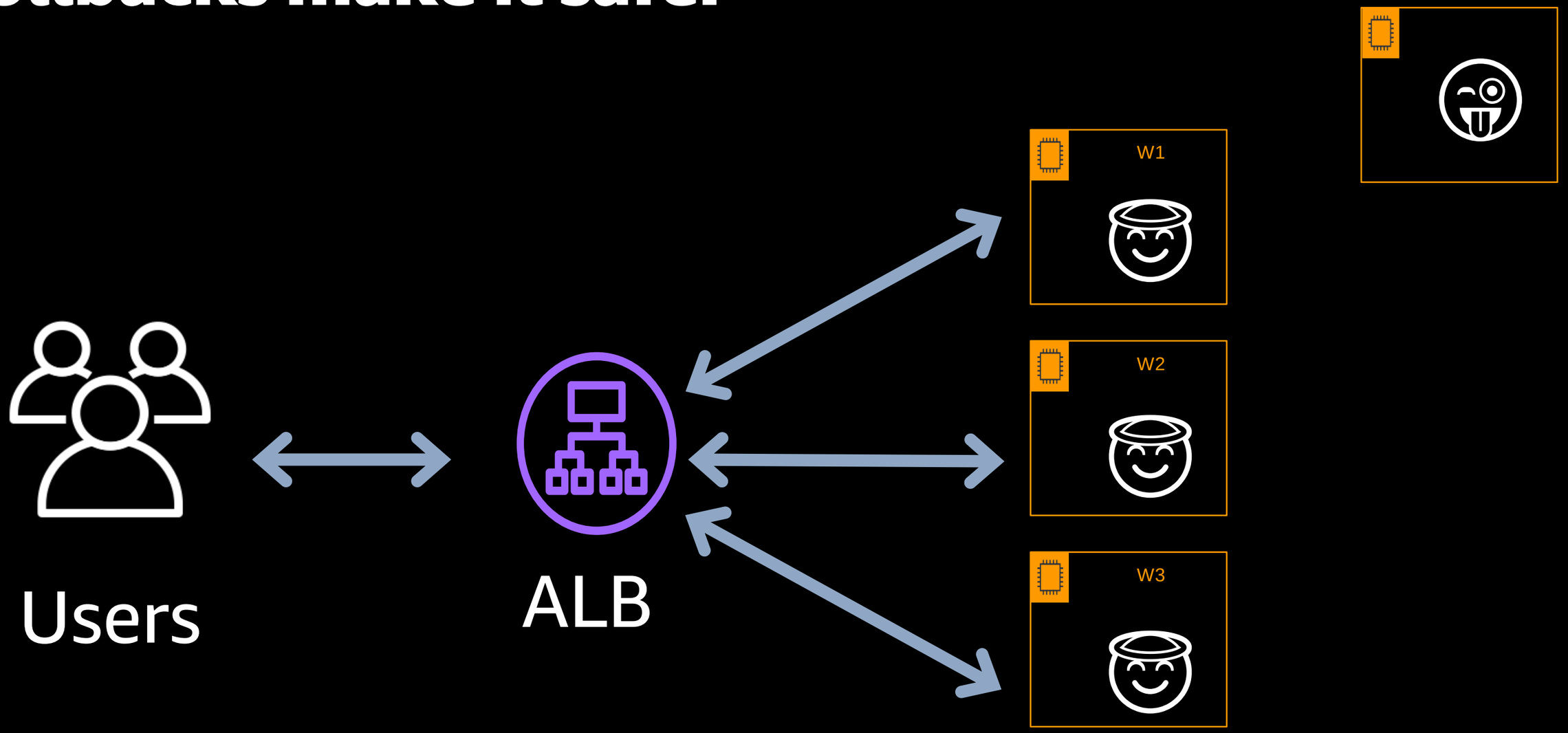
Placing the change inside the fleet



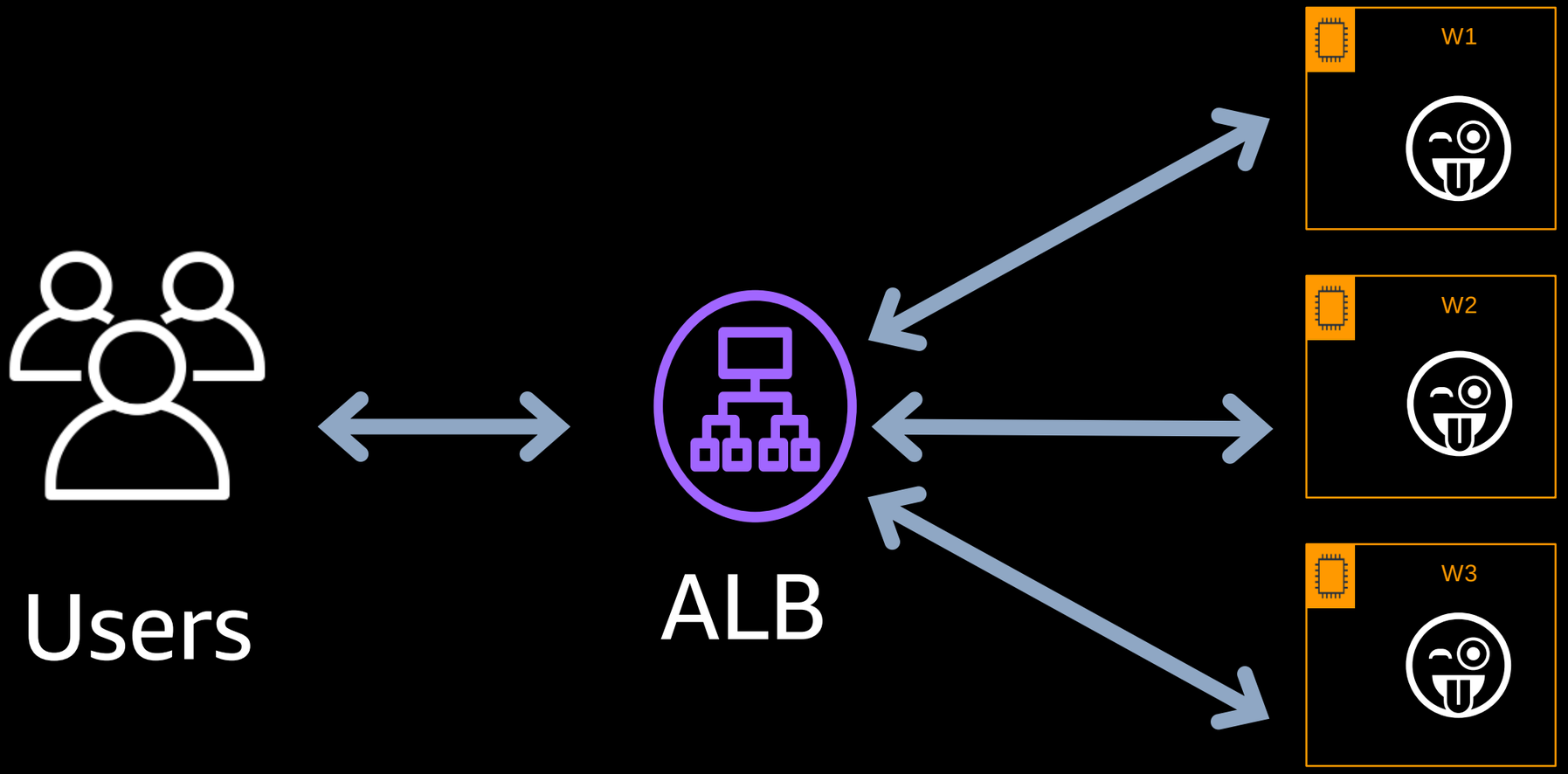
Wait and observe



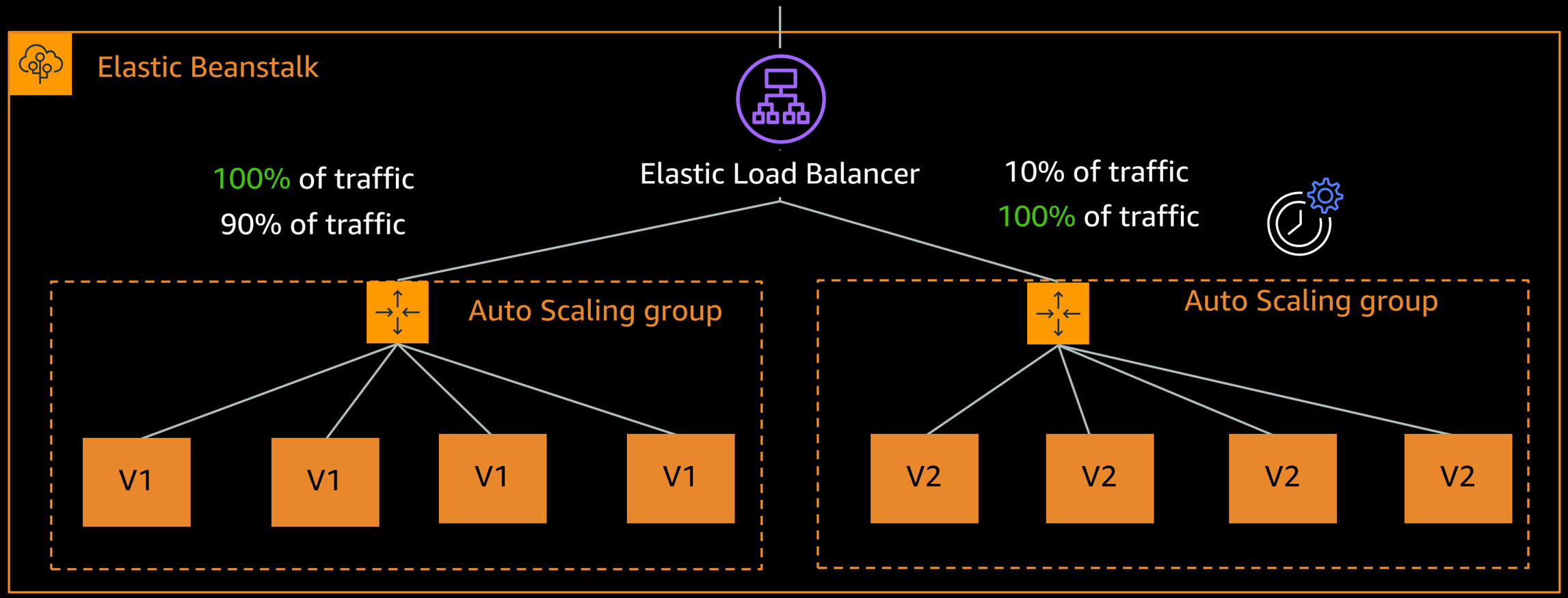
Rollbacks make it safer



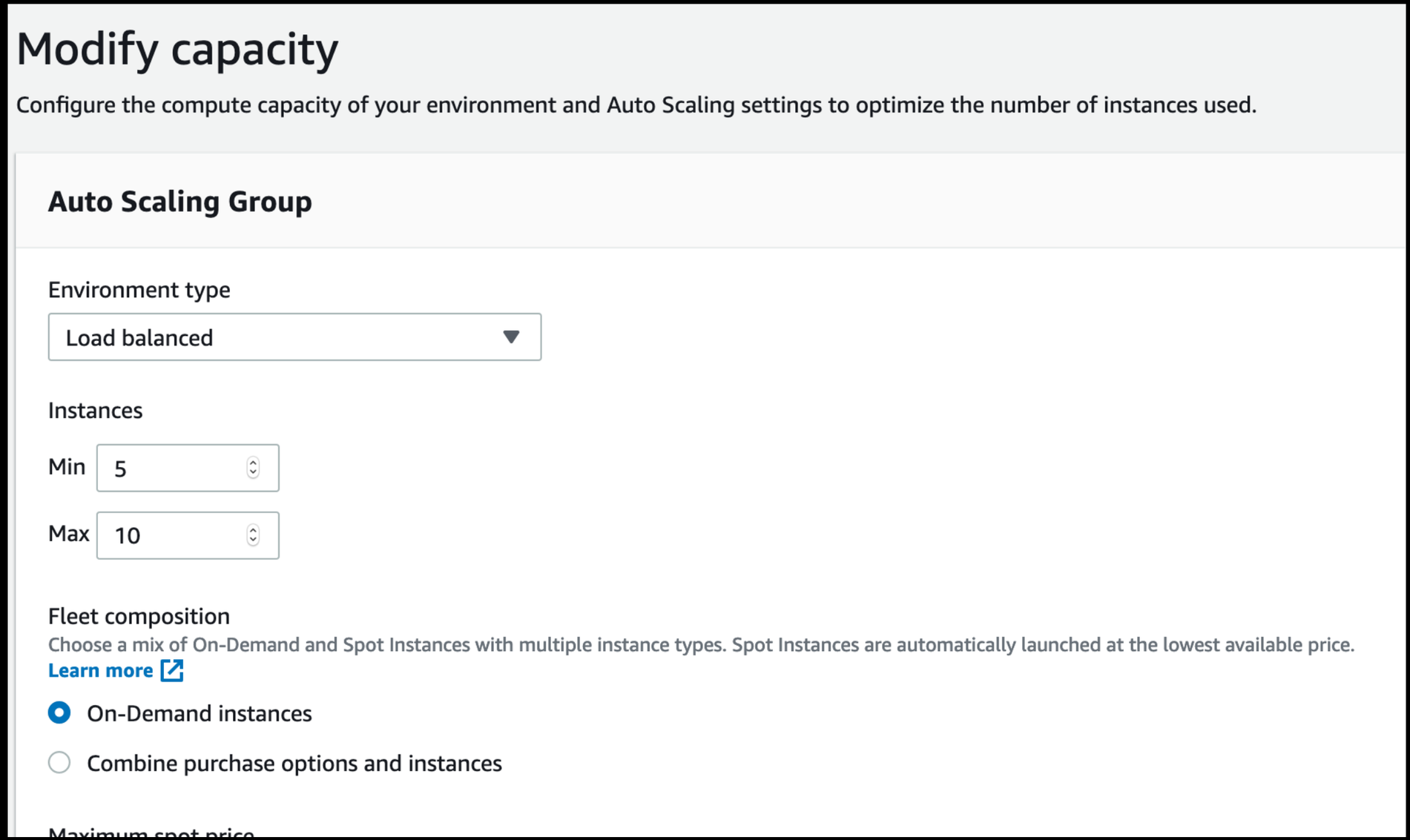
Fully promoted



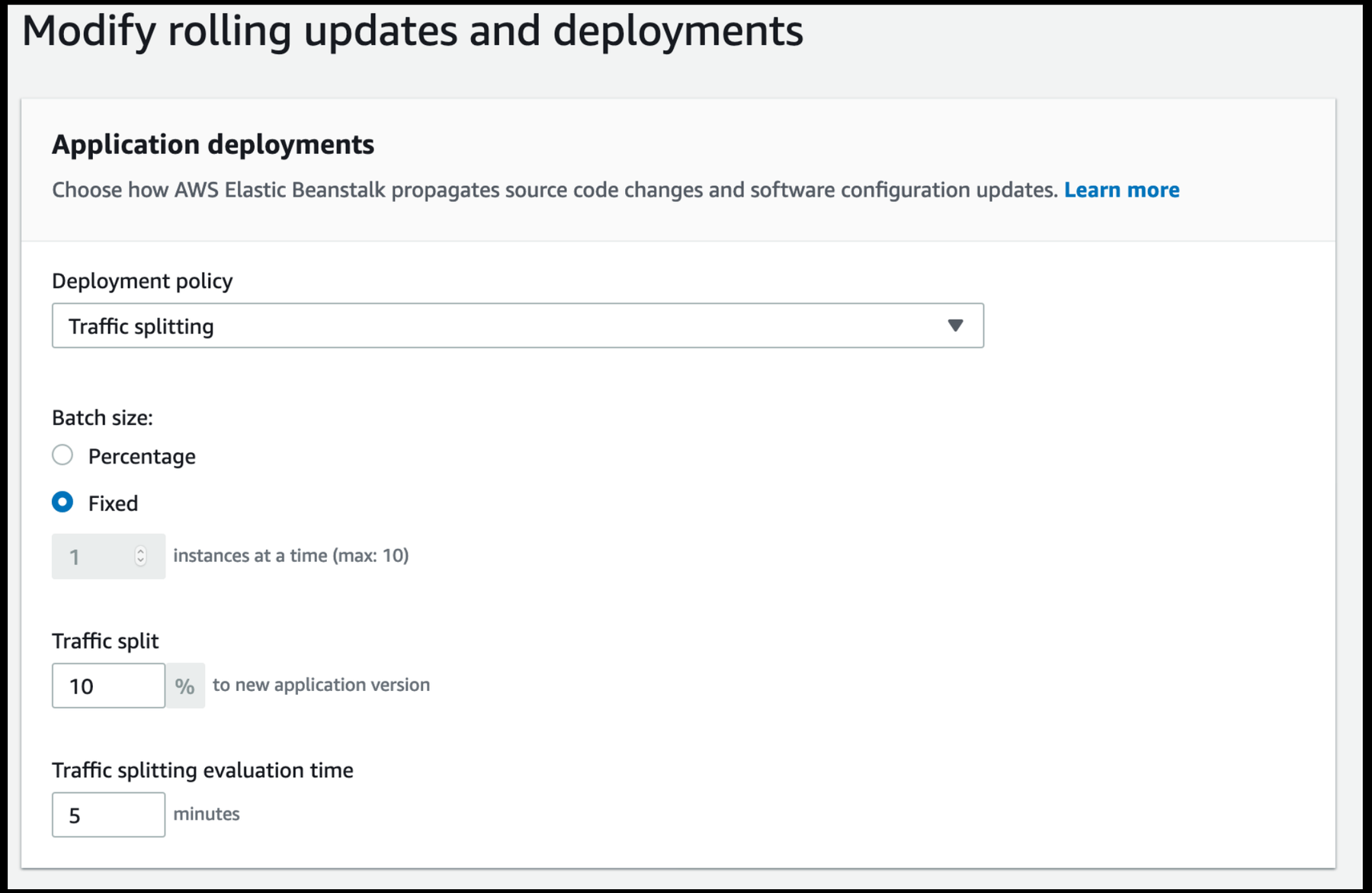
Implementing network traffic splitting



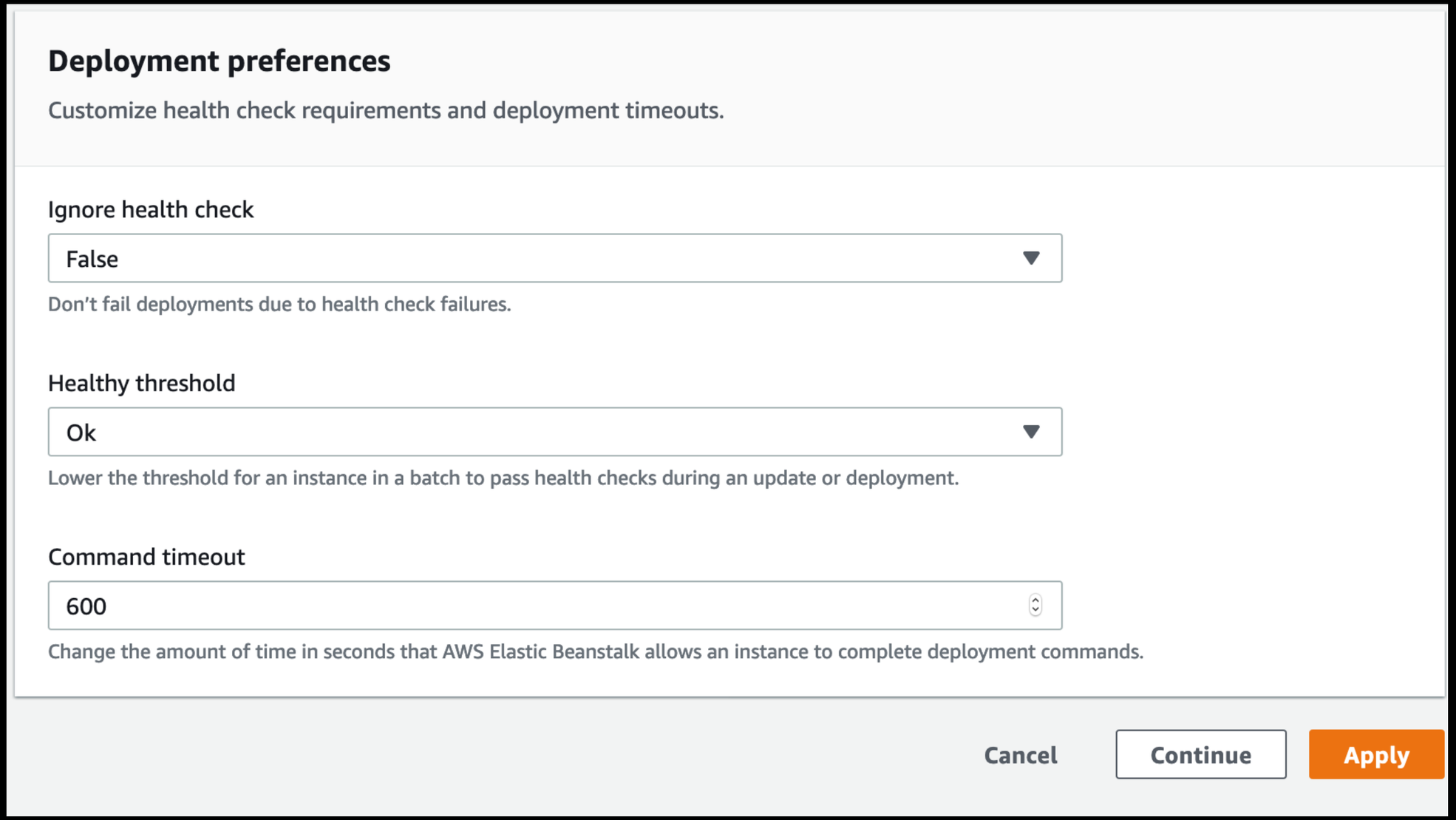
Setting a minimum number of instances



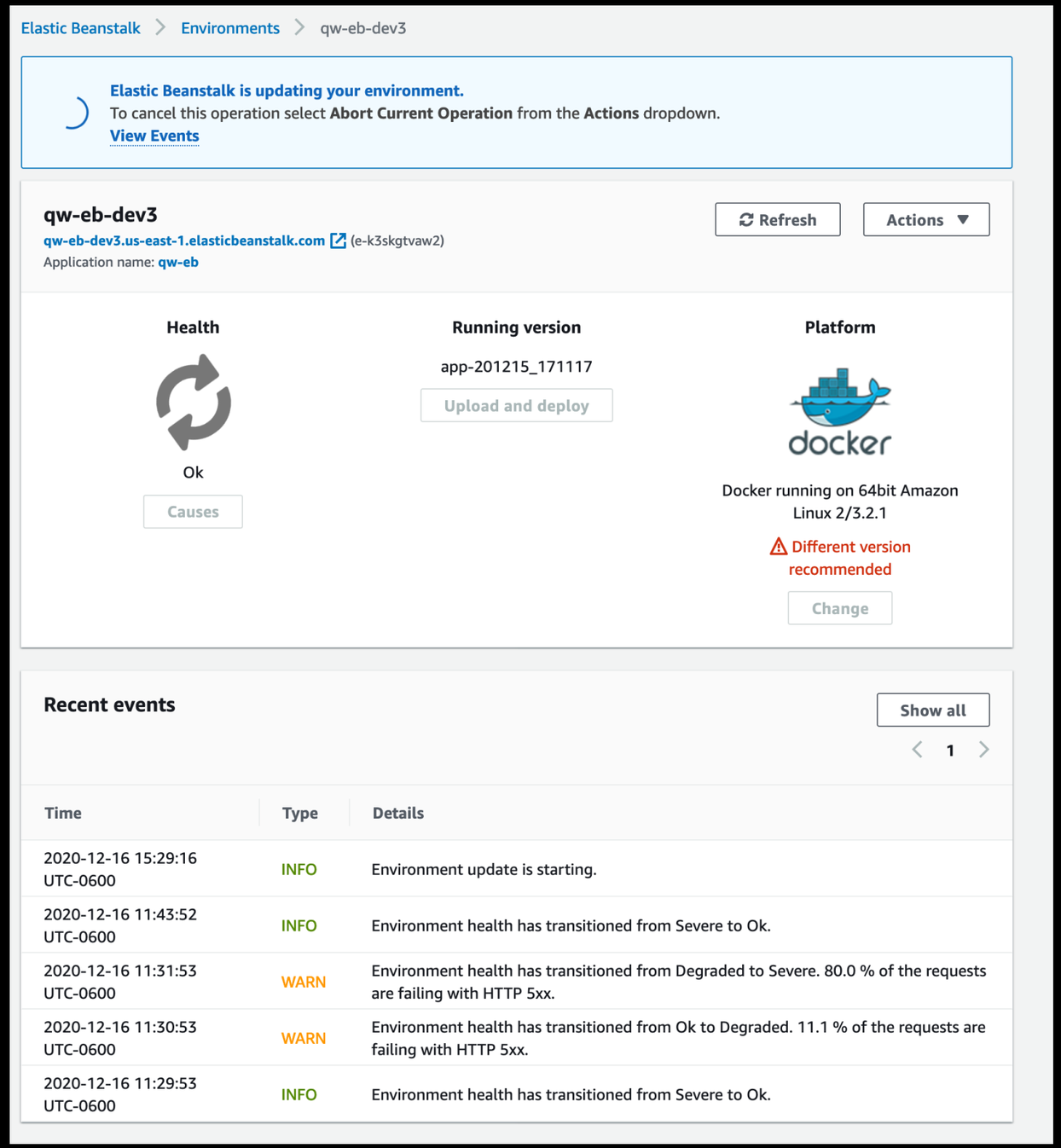
Setting the splitting policy



Additional preferences



Deployment in process



Traffic splitting confirmed!

Recent events

Show all

1

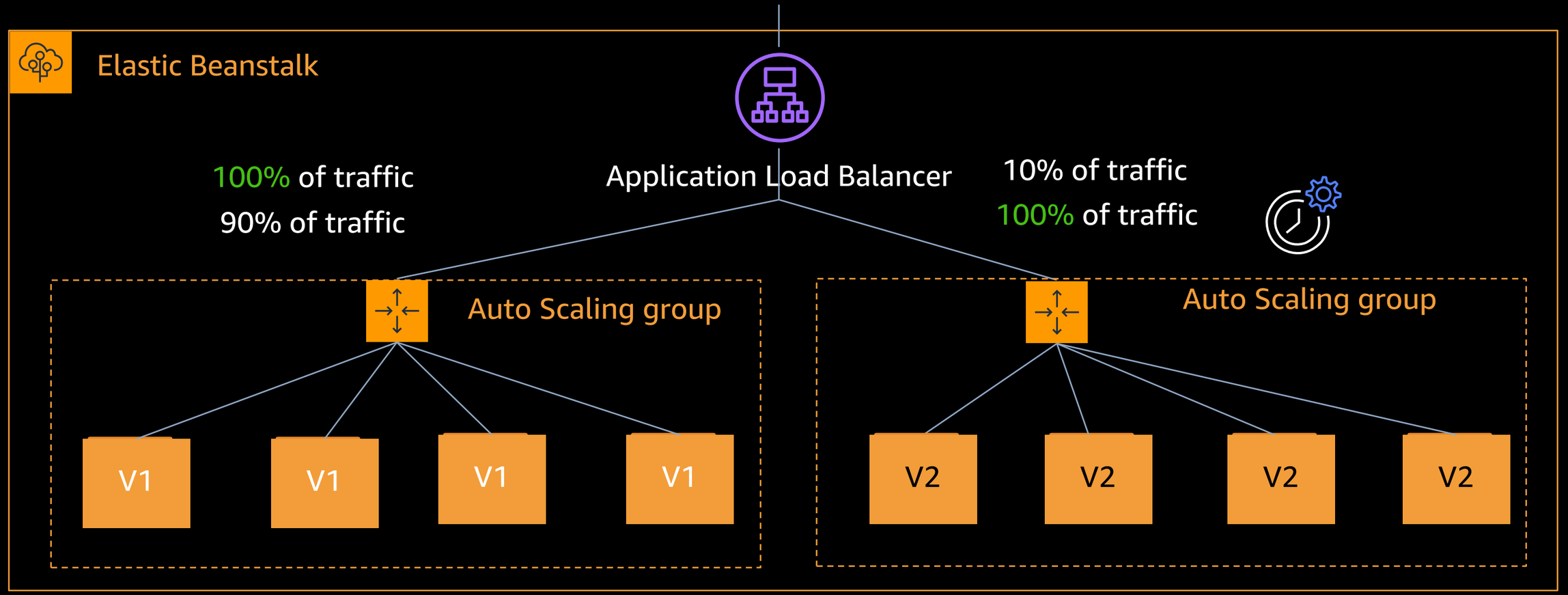
Time	Type	Details
2020-12-16 16:29:48 UTC-0600	INFO	Waiting for instance(s) [redacted] to pass health checks.
2020-12-16 16:29:48 UTC-0600	INFO	Traffic-splitting deployment: starting evaluation time for 20 minute(s).
2020-12-16 16:29:48 UTC-0600	INFO	Traffic-splitting deployment: routed 20% of client traffic to the new application version.
2020-12-16 16:29:48 UTC-0600	INFO	Traffic-splitting deployment: starting to shift 20% of client traffic to the new application version.
2020-12-16 16:29:48 UTC-0600	INFO	Traffic-splitting deployment: all instances that have the new application version passed load balancer health checks.

Both fleets running in parallel

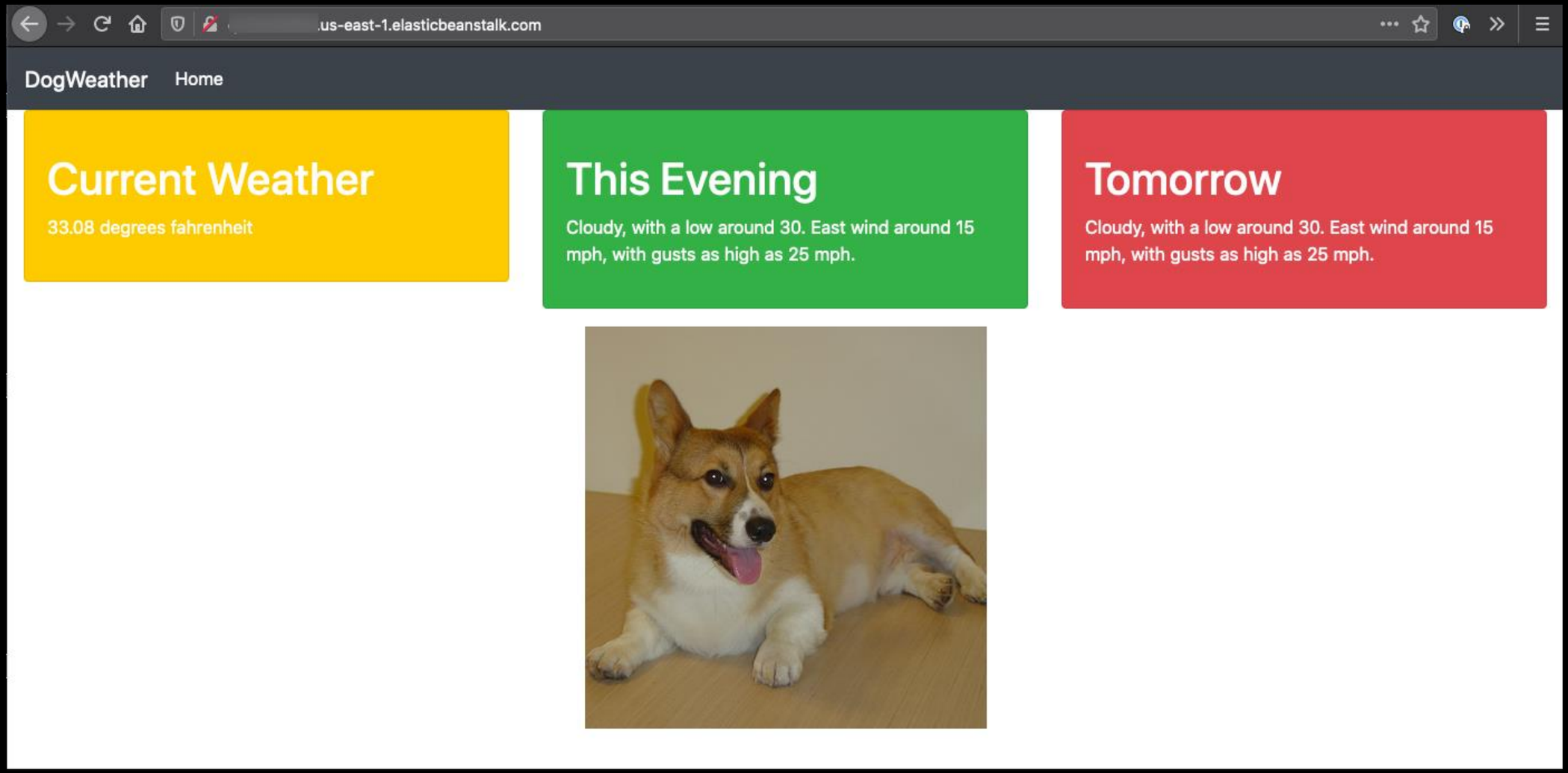
	Instance ID	Status	Running	Deployment ID
	Overall	Info • Application update in progress on 5 instances. 0 out of 5 instances completed (running for 4 minutes).	N/A	N/A
	[redacted]	Ok	4 minutes	7
	[redacted]	Ok	4 minutes	7
	[redacted]	Ok	4 minutes	7
	[redacted]	Ok	4 minutes	7
	[redacted]	Ok	4 minutes	7
	[redacted]	Ok	1 day	6
	[redacted]	Ok	1 day	6
	[redacted]	Ok	1 day	6
	[redacted]	Ok	1 day	6
	[redacted]	Ok	28 days	6



Let’s go back; see both fleets now?



Rolled out!


.us-east-1.elasticbeanstalk.com
DogWeather
Home
Current Weather
33.08 degrees fahrenheit
This Evening
Cloudy, with a low around 30. East wind around 15 mph, with gusts as high as 25 mph.
Tomorrow
Cloudy, with a low around 30. East wind around 15 mph, with gusts as high as 25 mph.

Elastic Beanstalk shared load balancing



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scenario: Releasing DogWeather with an API



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DogWeather? More like CatWeather!

- Our app is popular
- Other teams want to have an API for the service
- But we want to keep the core web service secure

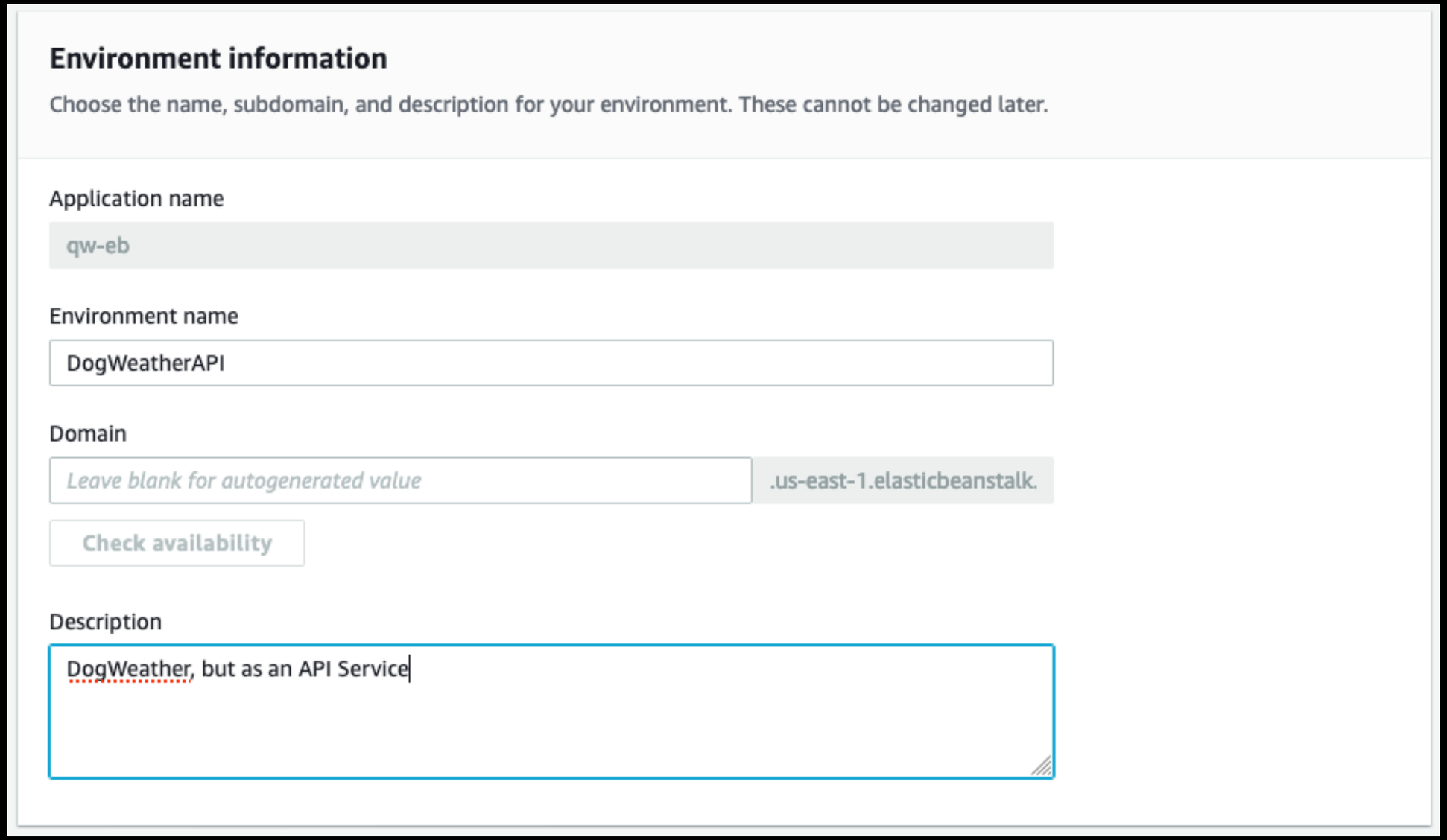
- We'll provide connectivity via a shared ALB
- Create a new, separate environment for this change
- Hook into an existing load balancer

Implement our API

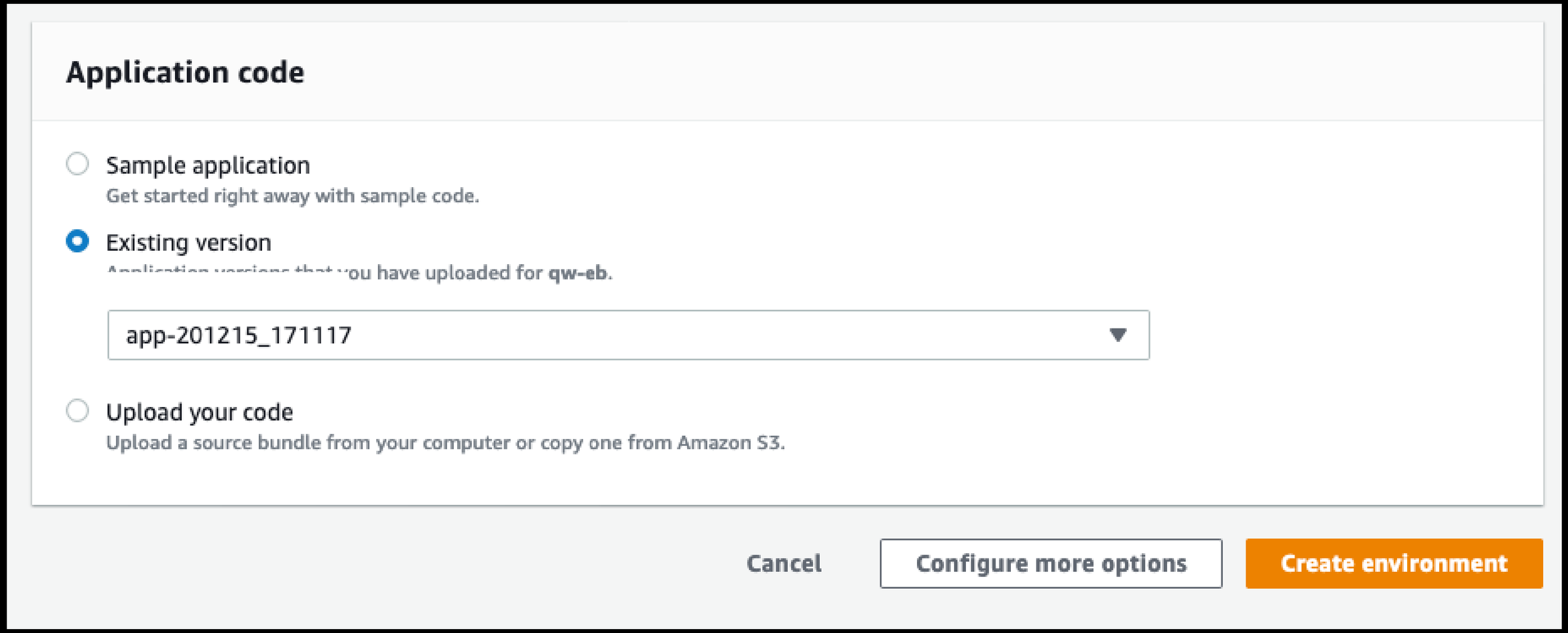
```python
@bp.route('/api/v1/current_weather')
def index():
    local_weather = LocalForecast(60614)
    # now this returns a dictionary of two values
    current_weather = local_weather.current()
    forecast = current_weather['forecast'] or current_weather['tomorrow_forecast']
    # integer value
    try:
        ftemp = current_weather['forecast_temperature'] or current_weather['tomorrow_forecast']
        cur_temp = current_weather['temperature']
        tom_temp = current_weather['tomorrow_temp']
    except KeyError:
        ftemp = 100
        cur_temp = 100
        tom_temp = 100

    params = {
        "name": "DogWeather",
        "title": "Hey folks it's Webdog!",
        "forecast": forecast,
        "temperature": cur_temp,
        "tomorrow_forecast": forecast,
        "tomorrow_temperature": tom_temp
    }
    return params
```

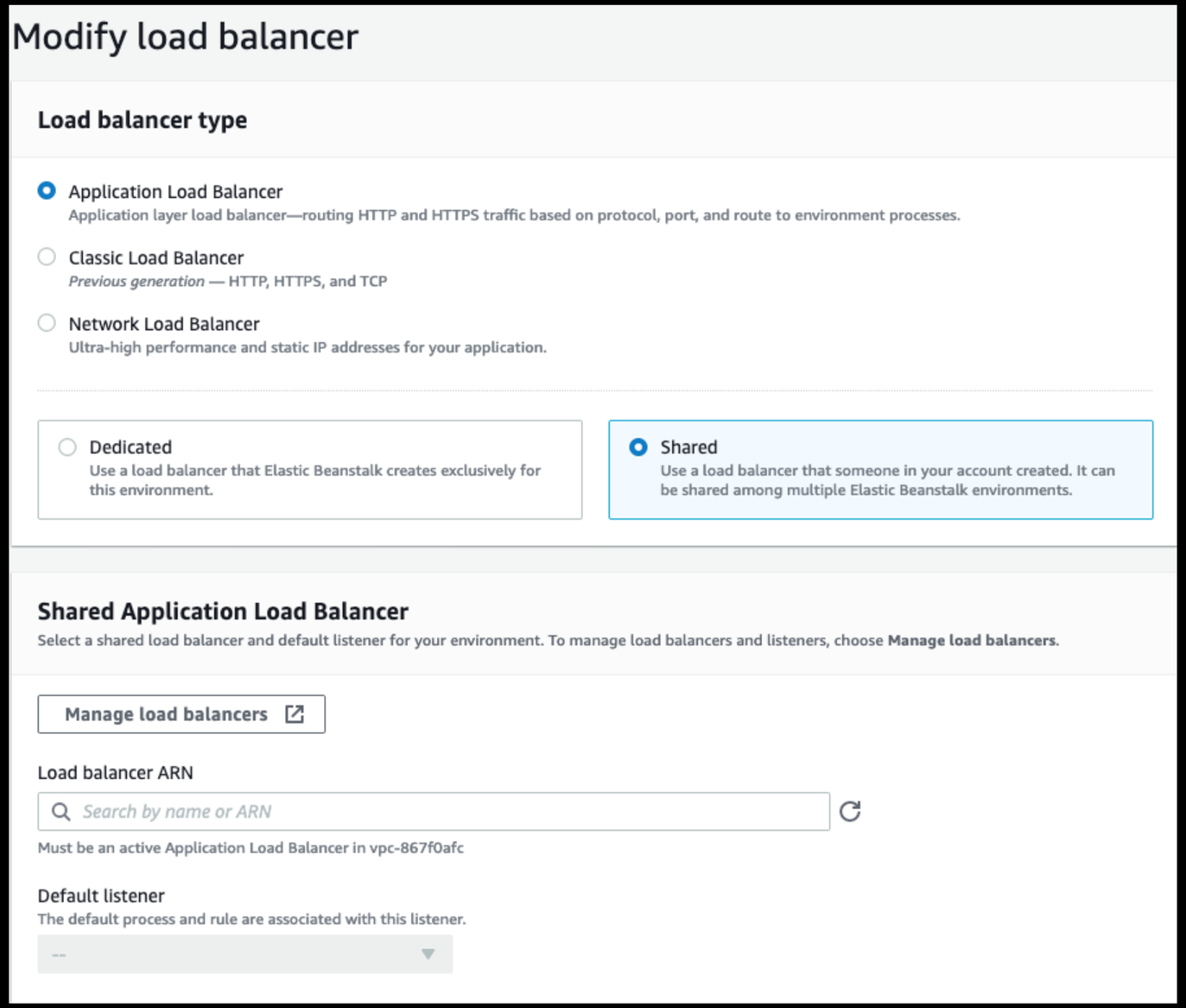
First, create a new environment from the app



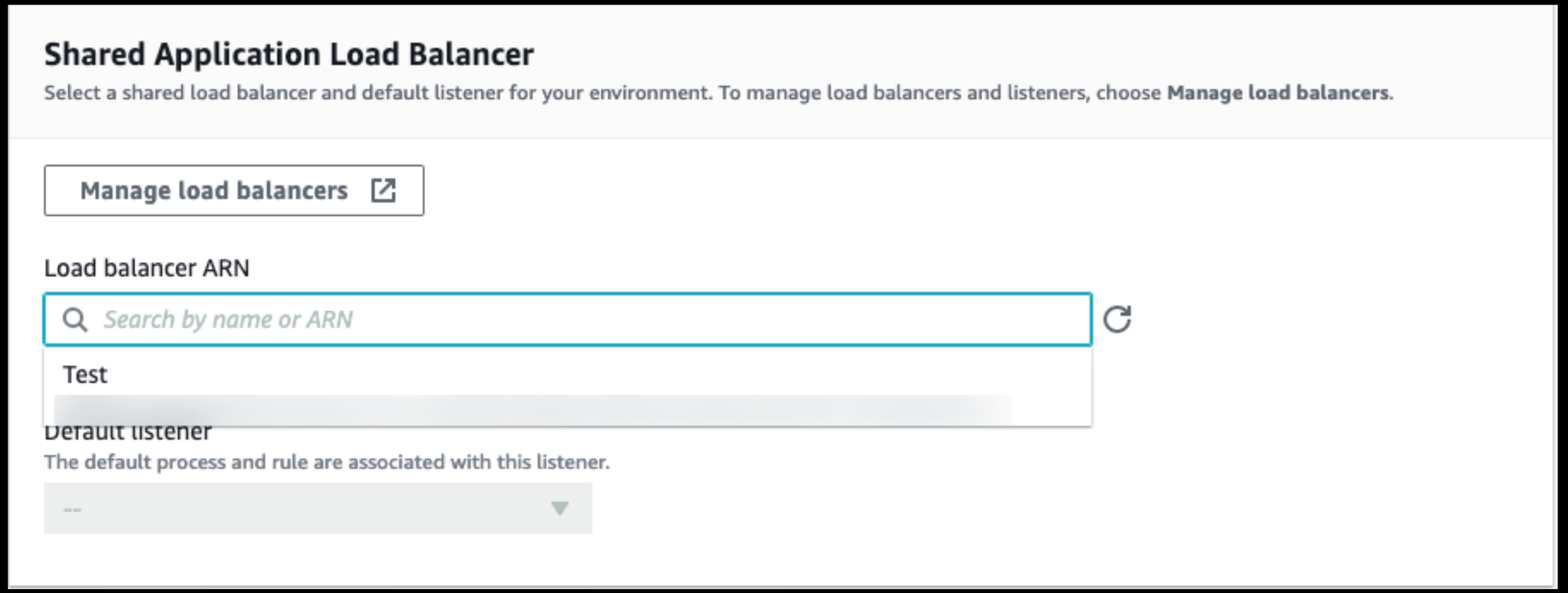
Select the latest version with the API implemented



Modify load balancer



Hooking in ...

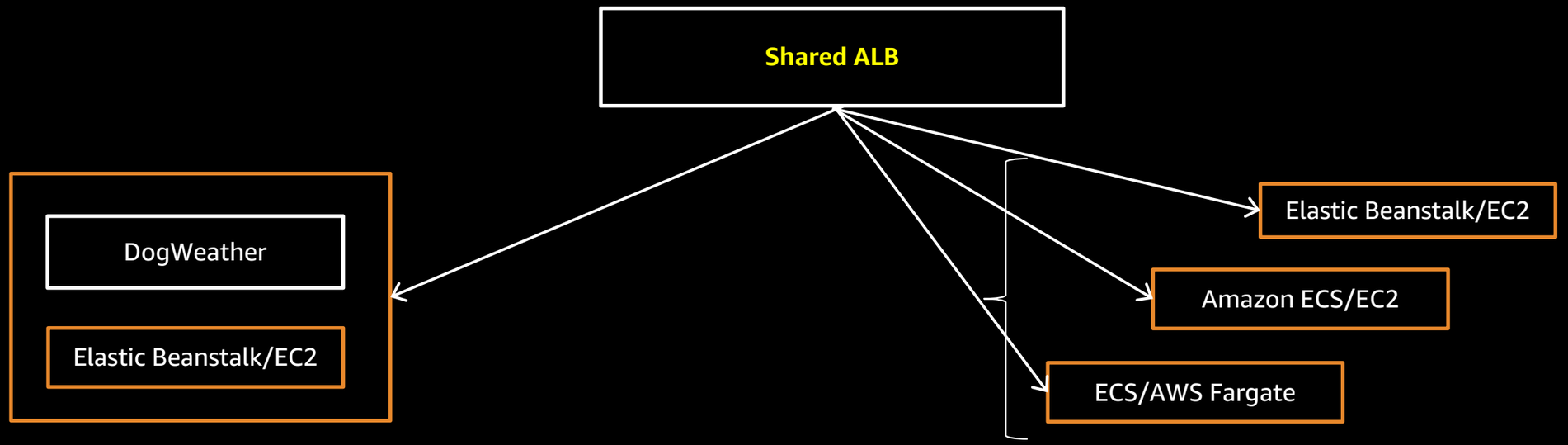

Shared Application Load Balancer
Select a shared load balancer and default listener for your environment. To manage load balancers and listeners, choose Manage load balancers.
Manage load balancers
Load balancer ARN
Search by name or ARN
Test
Default listener
The default process and rule are associated with this listener.
--

And requesting to the shared ALB

```
❯ http http://super-secret-shared-alb-path/api/v1/current_weather
HTTP/1.1 200 OK
Connection: keep-alive
Content-Encoding: gzip
Content-Type: application/json
Date: Wed, 16 Nov 1857 21:59:23 GMT
Server: nginx/1.18.0
Transfer-Encoding: chunked

{
    "forecast": "Cloudy, with a low around 30. East wind around 15 mph, with gusts as high as 25 mph.",
    "name": "DogWeather",
    "temperature": 33.08,
    "title": "Hey folks it's Webdog!",
    "tomorrow_forecast": "Cloudy, with a low around 30. East wind around 15 mph, with gusts as high as 25 mph.",
}
```

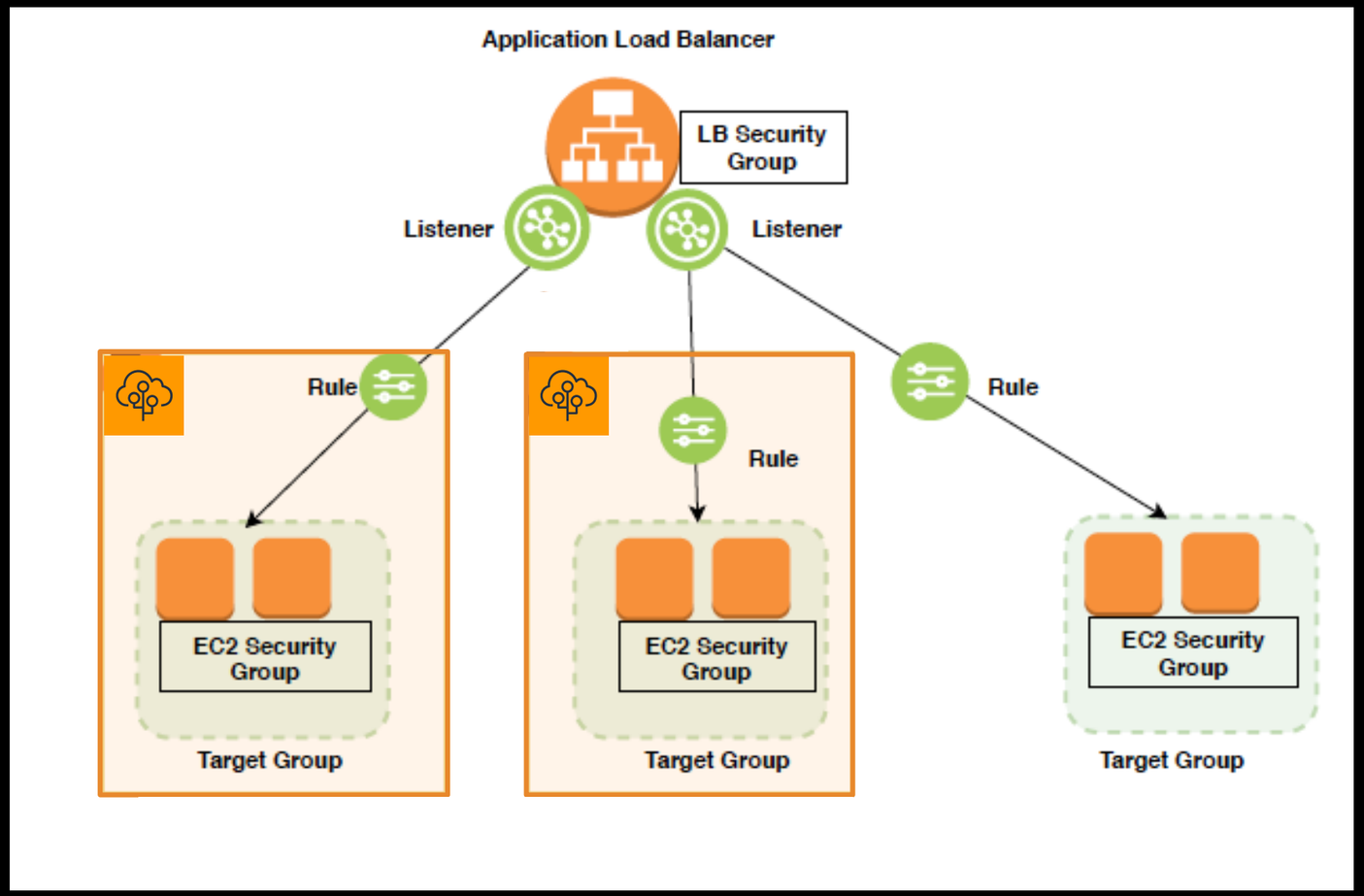
What you've created



- ✓ Decoupled components can scale independently
- ✓ Promote distributed architectures
- ✓ Cost savings from sharing an ALB across multiple environments

Shared Application Load Balancer capability

- Share an ALB among Elastic Beanstalk environments
- Reduce ALB costs while modernizing your app
- Use **host-based** and **path-based** rules to serve traffic to multiple environments or services hosted on Elastic Beanstalk or Amazon ECS





Thank you!

Christian Weber

Twitter: @classicwebdog

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.