
AWS
re:Invent

EMB008

# AWS Proton: Automating infrastructure provisioning & code deployments

Rafa Alvarez

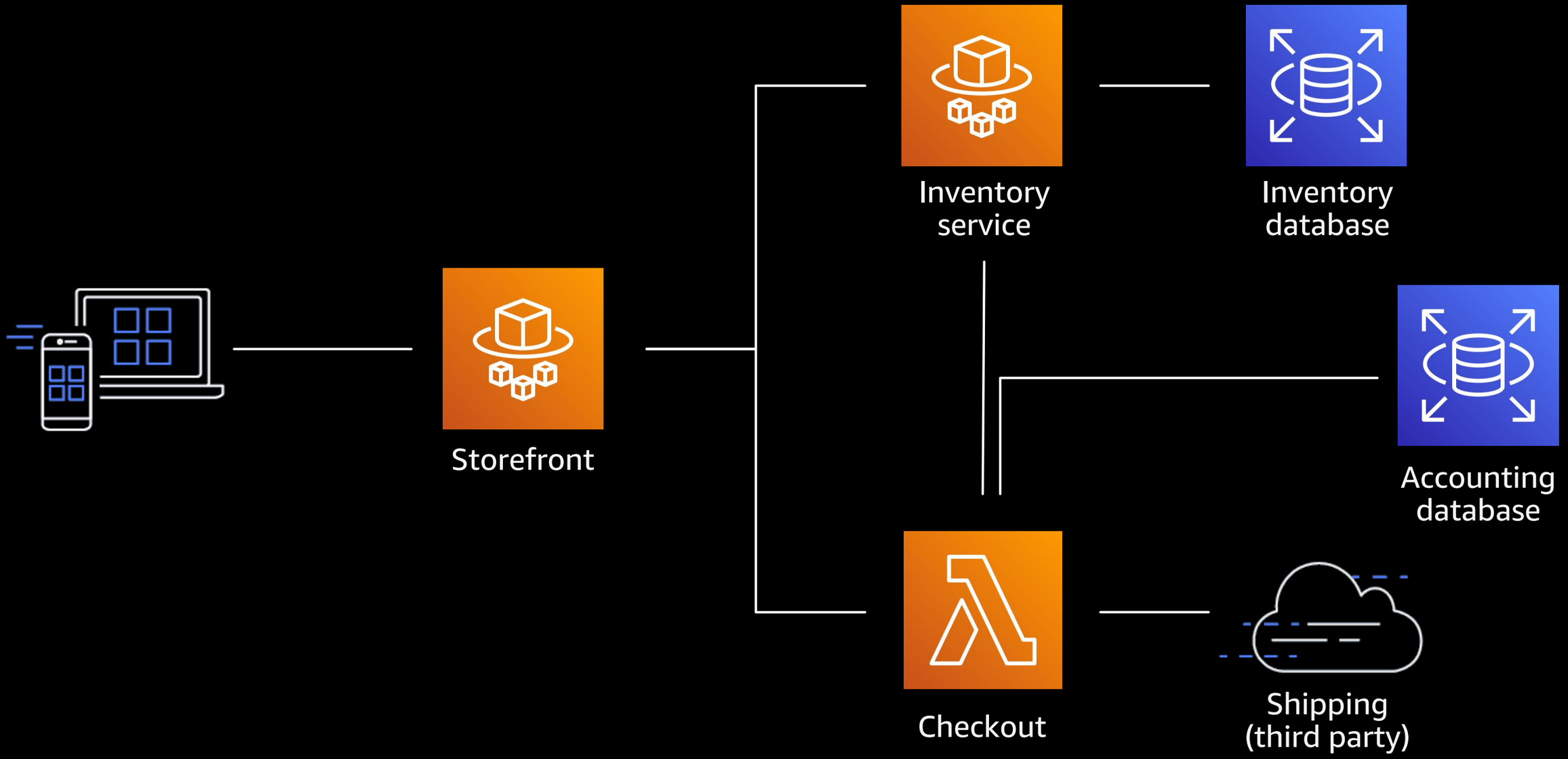
Sr. Product Manager

AWS

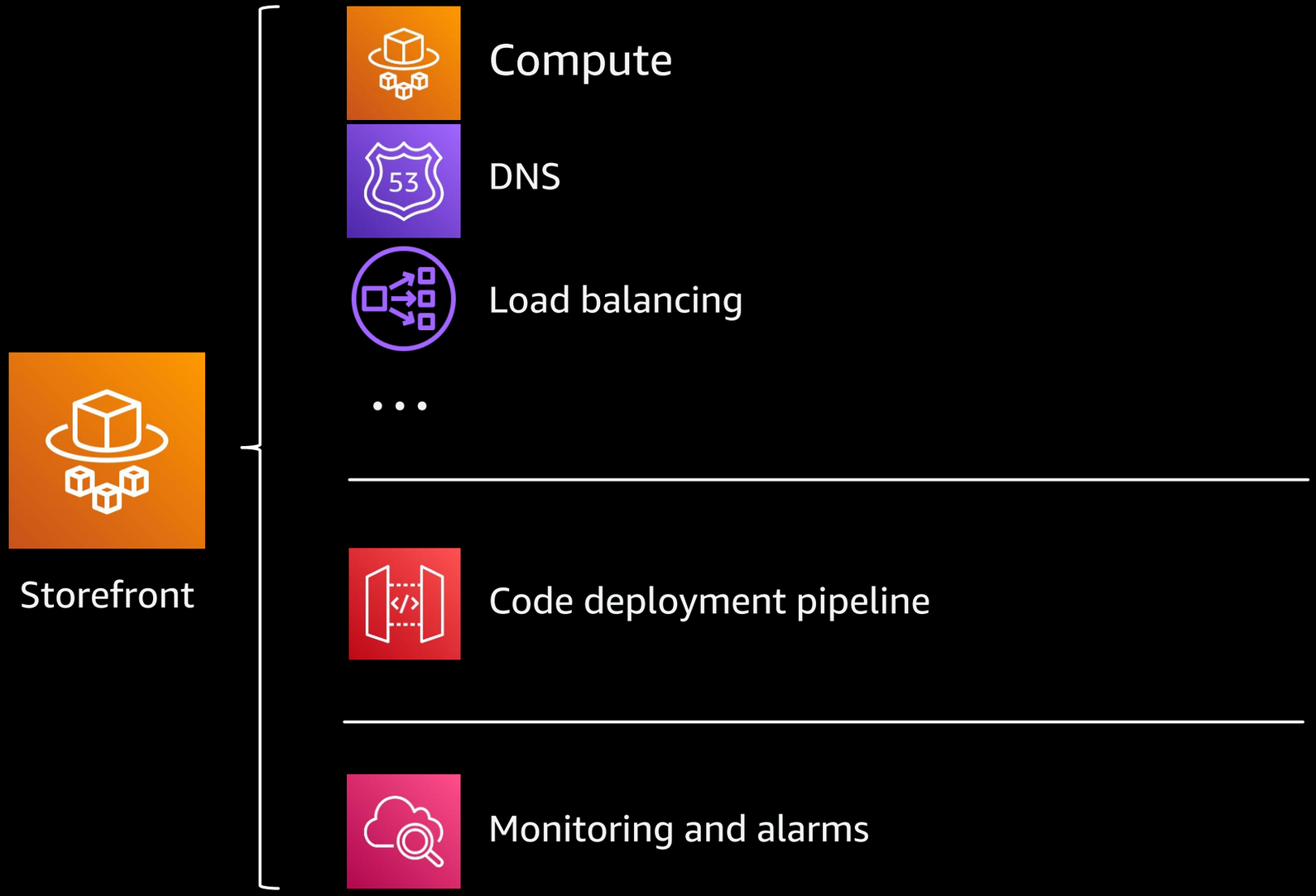
AWS re:Invent

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

# Using microservices for speed and agility



# “Simple” is not so simple

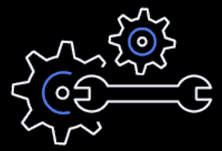


# Standardization vs. speed

## Infrastructure administrators – standards and management

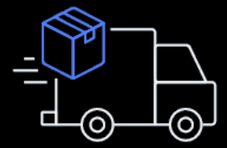


Enforce standards



Maintain infrastructure and keep it updated

## Development teams – speed and autonomy

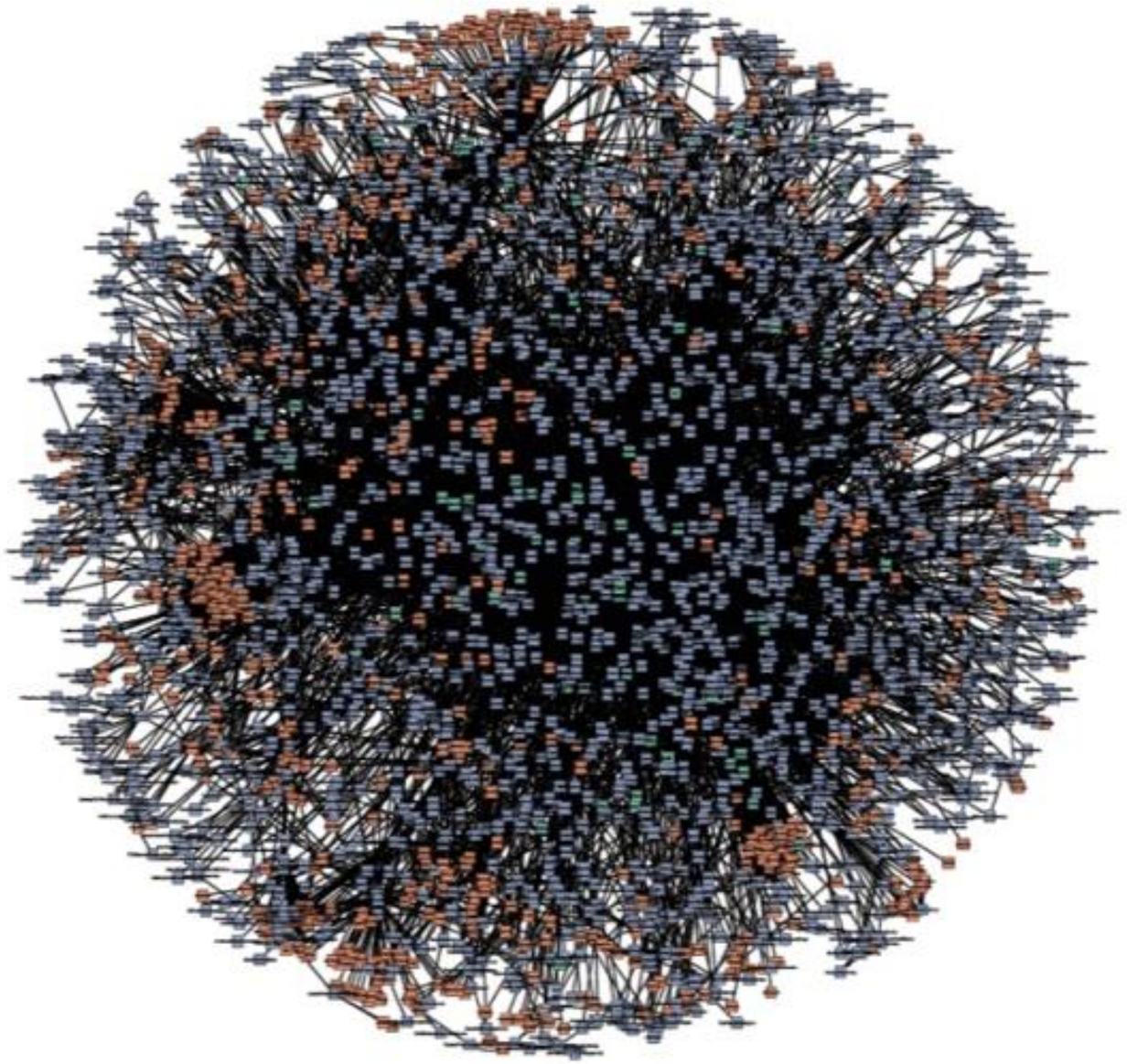


Self-service infrastructure and deployment

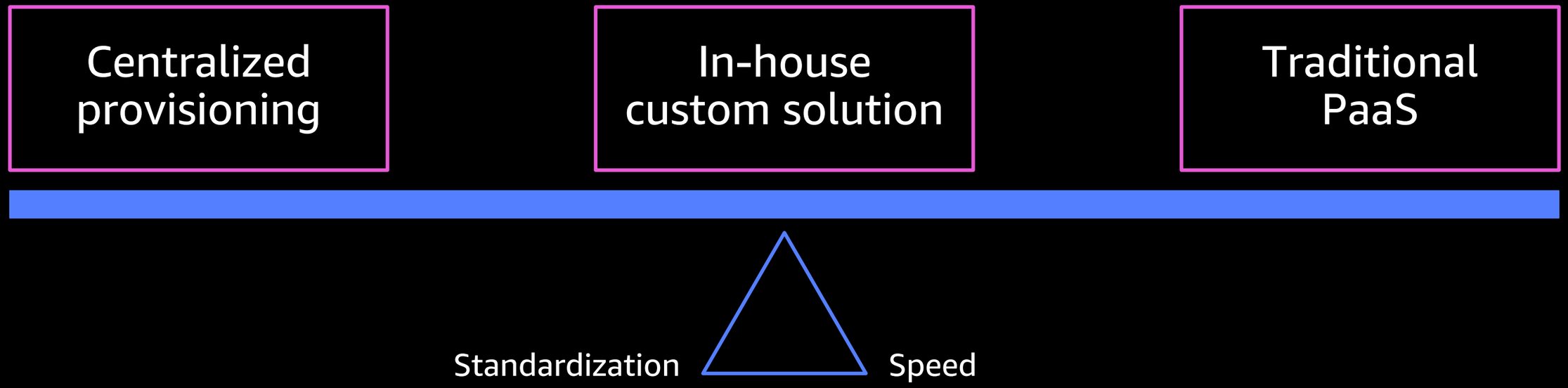


Transparent updates

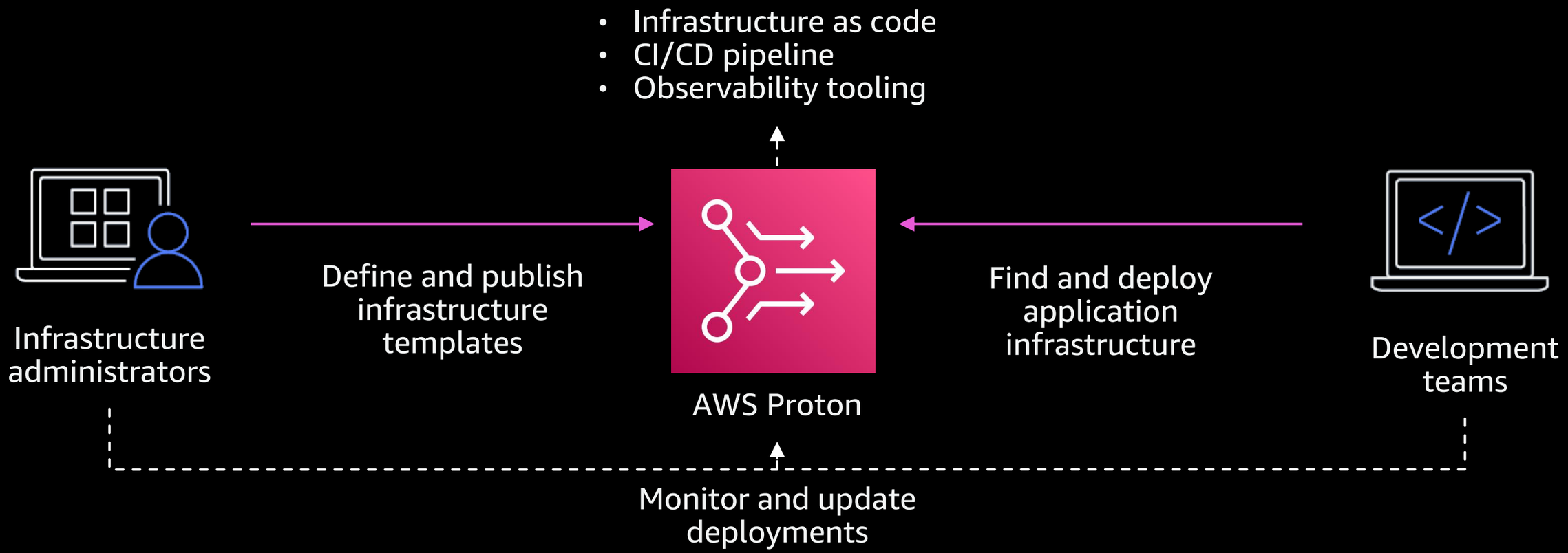
# What happens when the architecture grows?



# How do we bridge the gap?



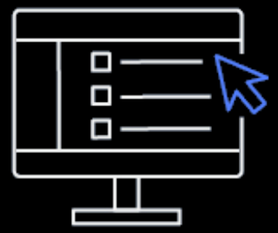
# AWS Proton: An application delivery service



# Core features



Self-service infrastructure and code deployment for developers



Centralized infrastructure management and one-click upgrades



Integration with third-party services

# Vision for AWS Proton

Help customers quickly and confidently adopt **best practices** and **new features** in containers and serverless

Enable rapid, consistent adoption of **cloud-based modern application development** for customers

Allow infrastructure teams to focus on **deciding the best architecture for the business**, not operationalizing its usage

# AWS Proton concepts



**Environments**

Shared resources in which services can run

Examples: a VPC, a cluster

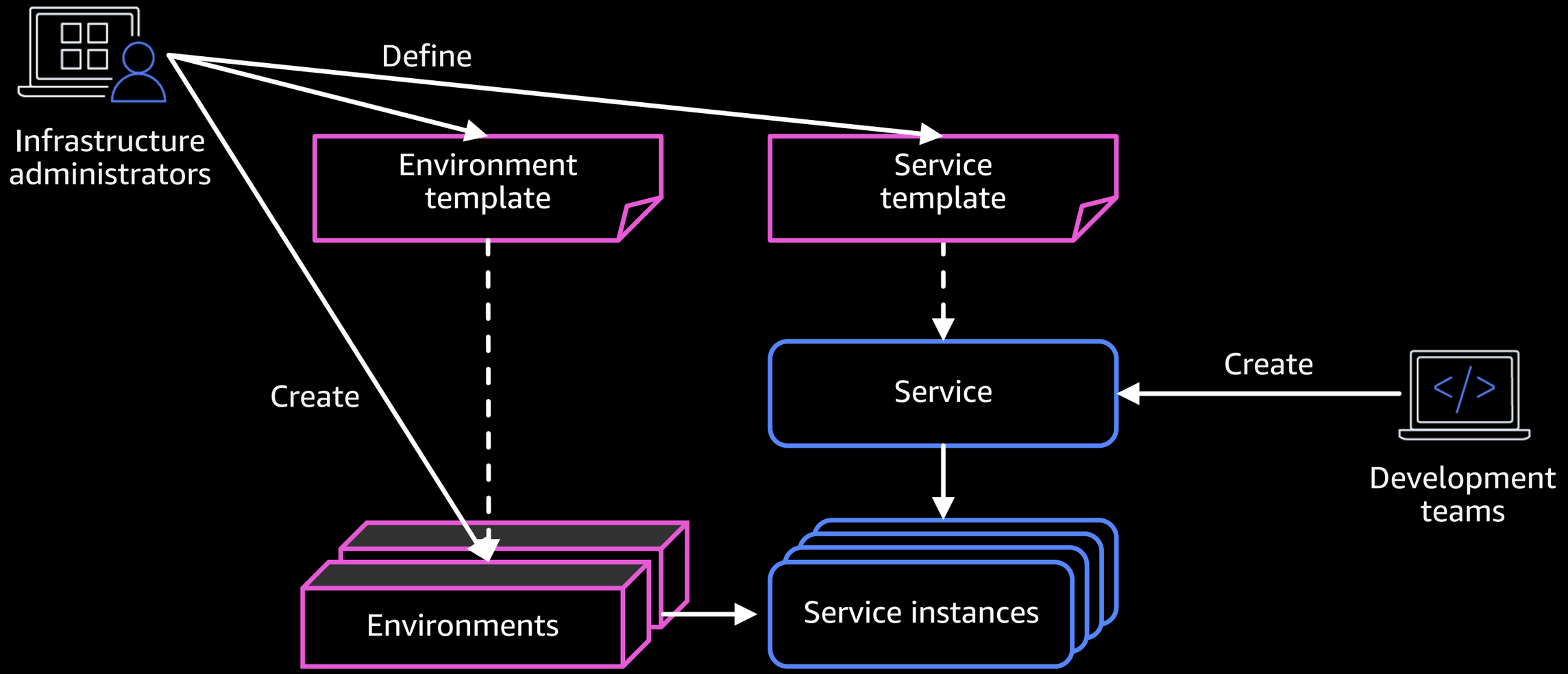


**Services**

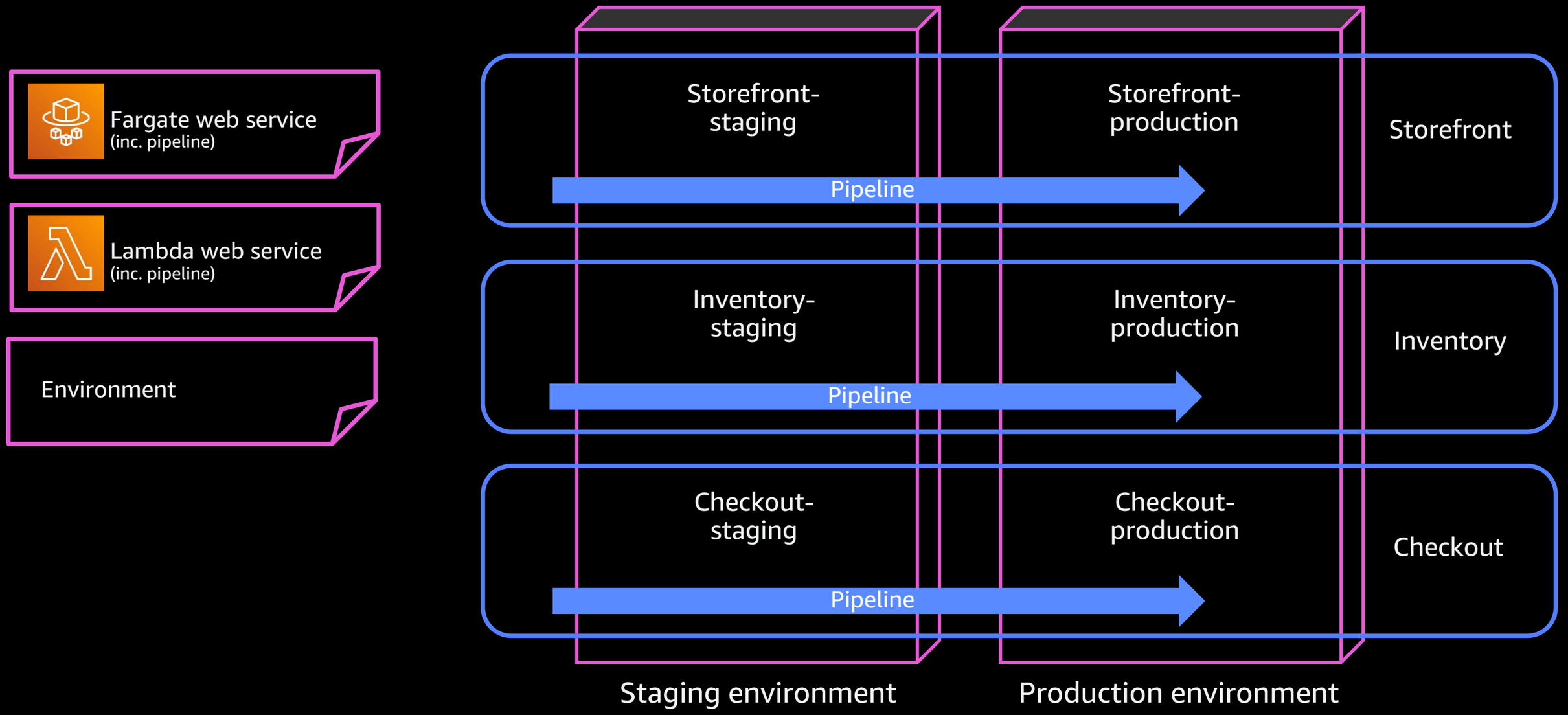
A workload running in an environment

Examples: an AWS Fargate web service, an AWS Lambda data processor

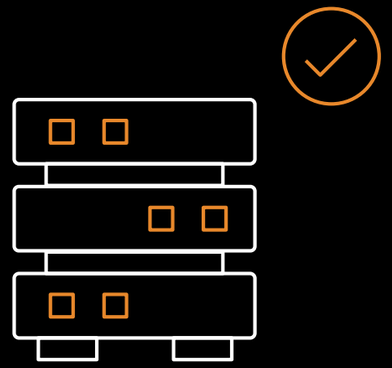
# Working with AWS Proton resources



# How does this map to our architecture?



# Administrators: Define standards

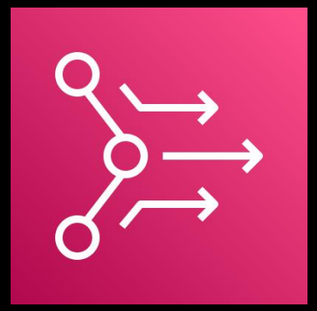


**Administrators**

Build and publish templates that model infrastructure, CI/CD pipelines, and observability tools

- Create templates
- Register templates in AWS Proton
- Manage template versions
- Deploy environments

# Anatomy of an AWS Proton template



- Parametrized infrastructure as code templates
- Parametrized CI/CD template
- AWS Proton schema
- Manifest

# Parametrizing an infrastructure as code template

AWS CloudFormation: Storefront-webservice.yaml

```
DNSRecord:
  Type: AWS:Route53:RecordSet
    Properties:
      Name:
        storefront.example.com
```

AWS CloudFormation: Webservice.yaml

```
DNSRecord:
  Type: AWS:Route53:RecordSet
    Properties:
      Name:
        {{subdomain}}.example.com
```

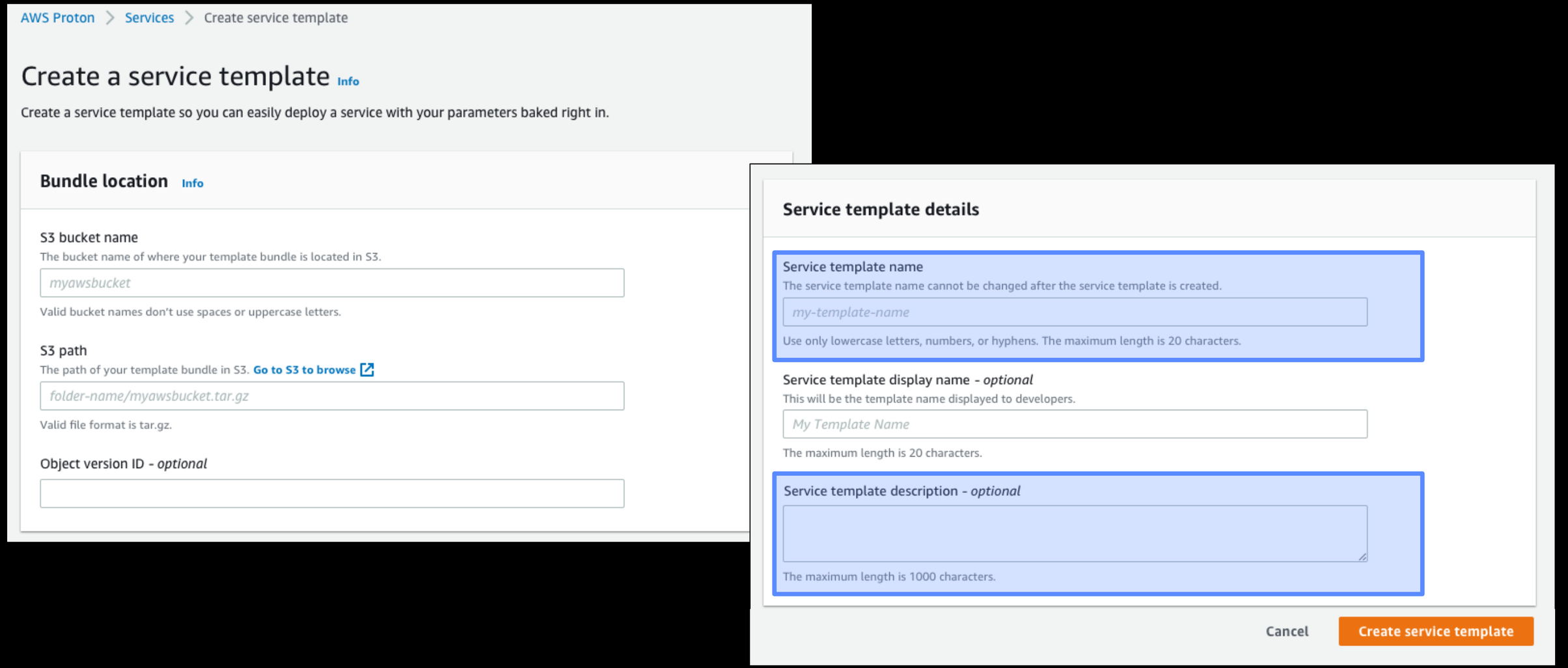
AWS CloudFormation: Storefront-staging-webservice.yaml

```
DNSRecord:
  Type: AWS:Route53:RecordSet
    Properties:
      Name:
        storefront-staging.example.com
```

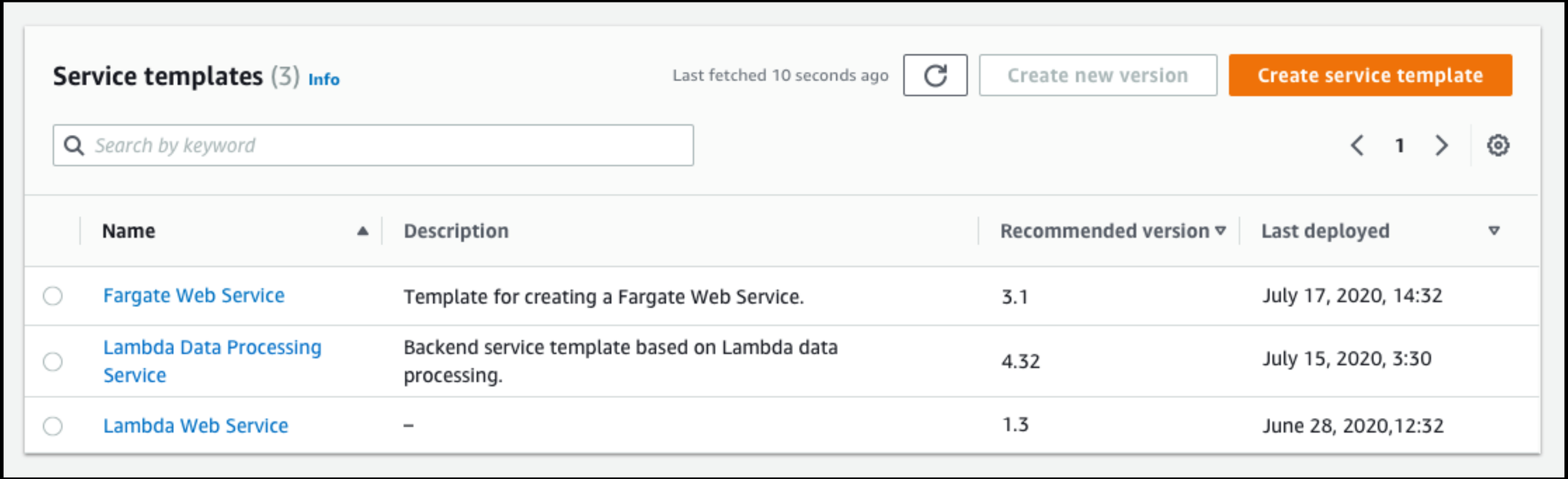
AWS Proton: Webservice-schema.yaml

```
ServiceInstanceInputs:
  type: object
  required:
    - subdomain
  subdomain:
    type: string
```

# Registering a template



# Managing your templates



# AWS Proton sample templates



## Well-Architected

Built using AWS Well-Architected best practices



## Open source

Customers can customize templates to suit their needs

**View on GitHub** – github.com/aws-samples/aws-proton-sample-templates

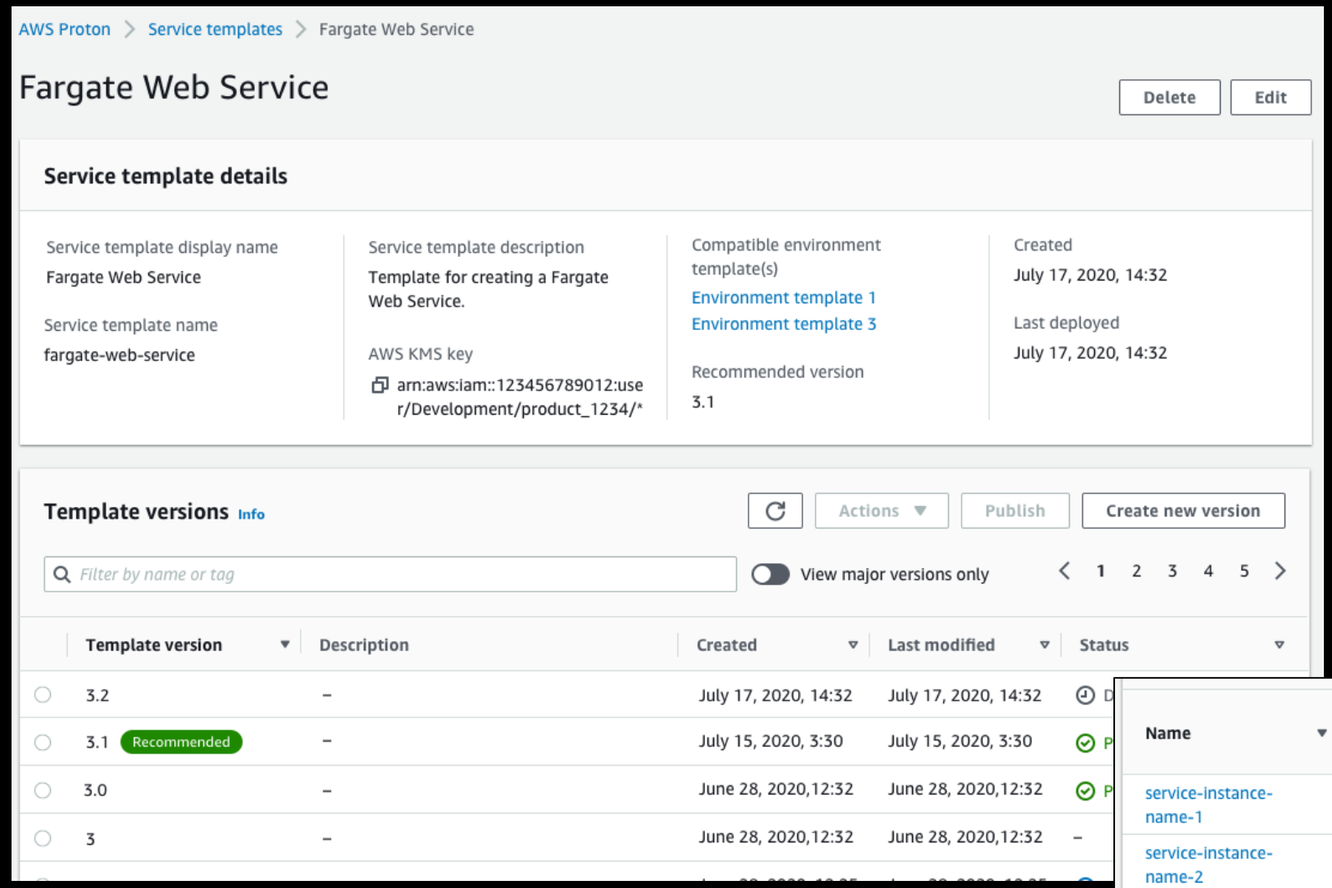
# Keeping infrastructure updated through template versions

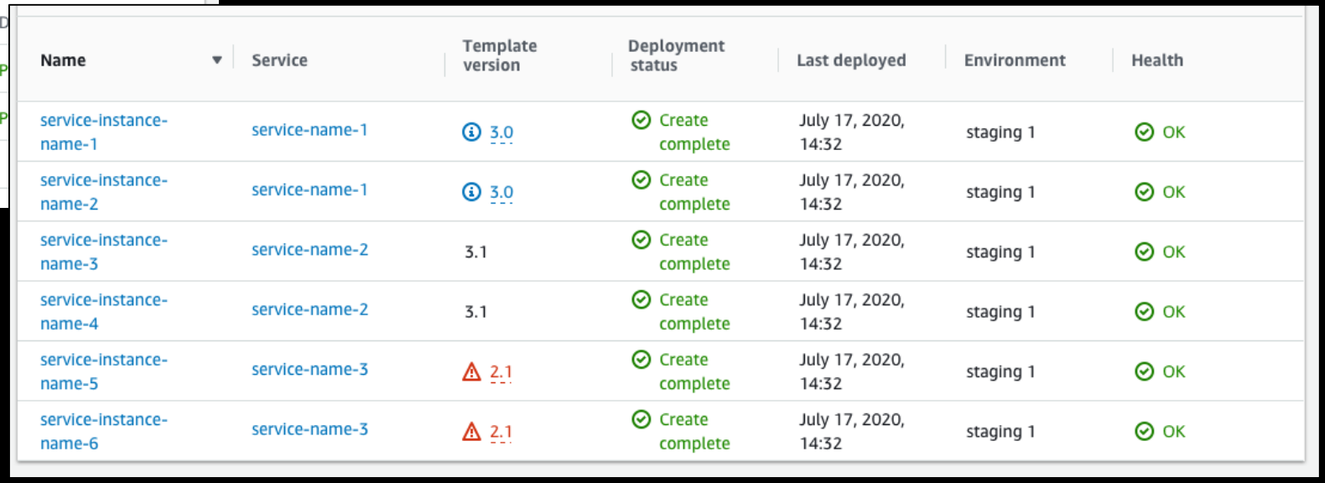
- New versions reflect changes in the template

- **Minor** updates (1.2 to 1.3): changes to the infrastructure
- **Major** updates (1.x to 2.x): changes to the schema (and inputs)

- Latest version: **recommended**

# Managing your template versions



# Environments

- **Shared resources** for multiple service instances

  “VPC with subnets”

- **Stages** in the pipeline
  - Production: VPC with public subnets
  - Staging: VPC with private subnets

- Include **IAM roles** for resource provisioning

# Developers: Code and deploy

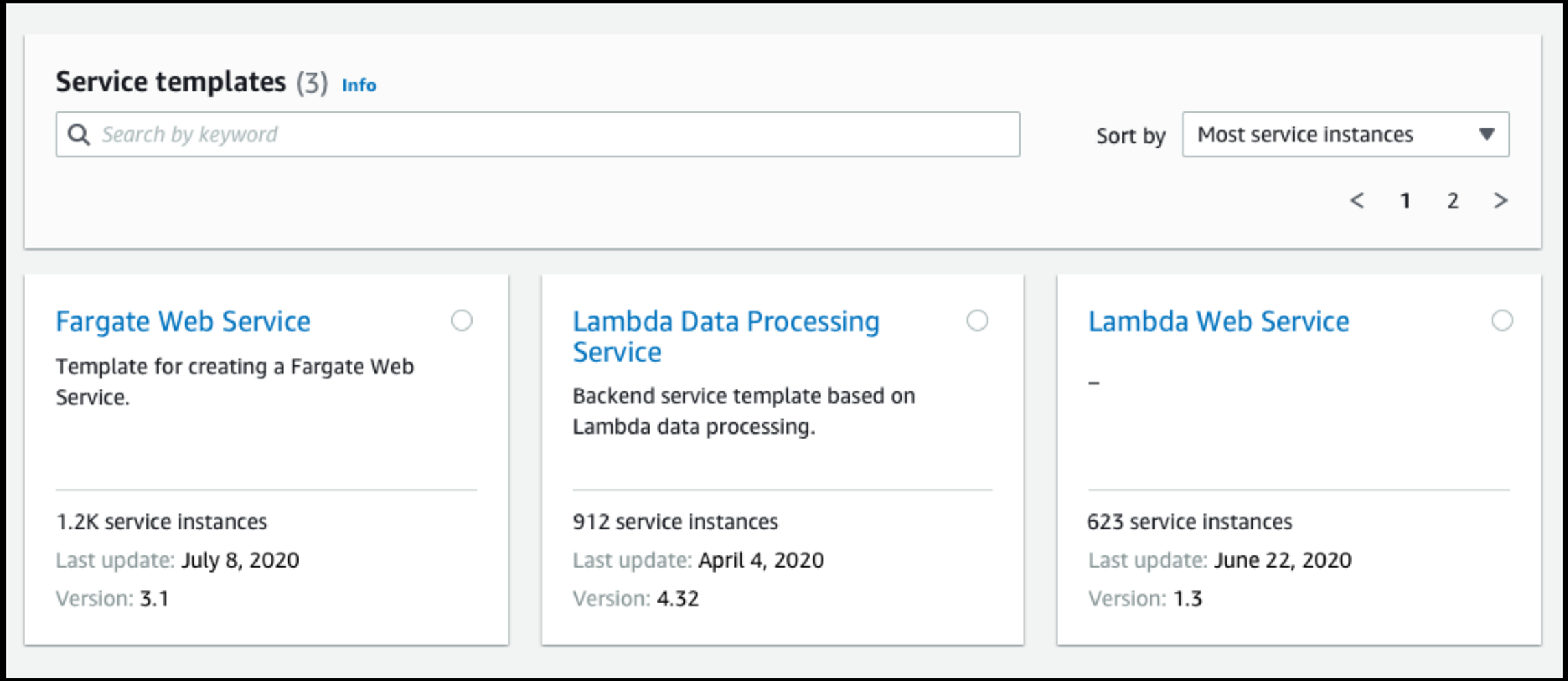


**Developers**

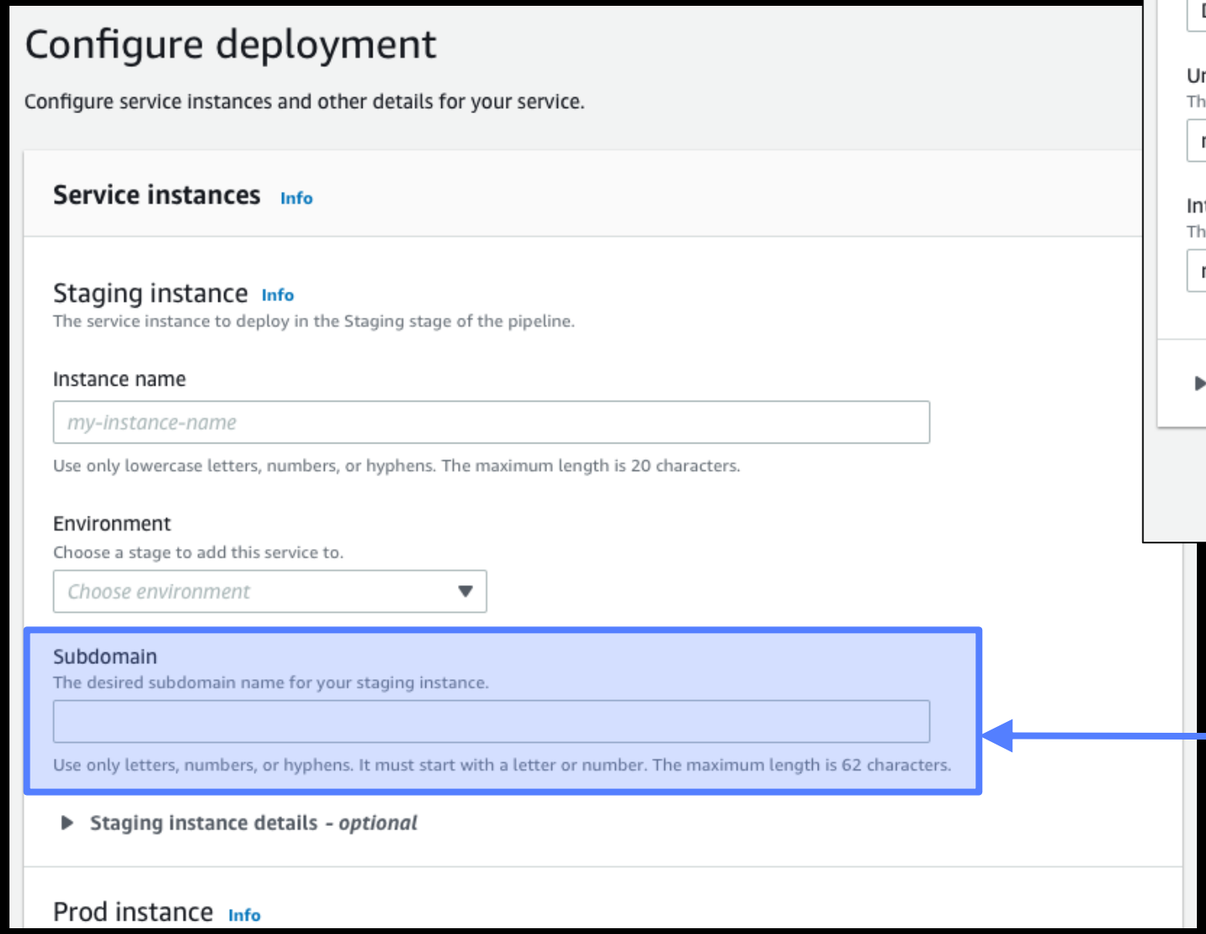
Find and deploy end-to-end services for your applications

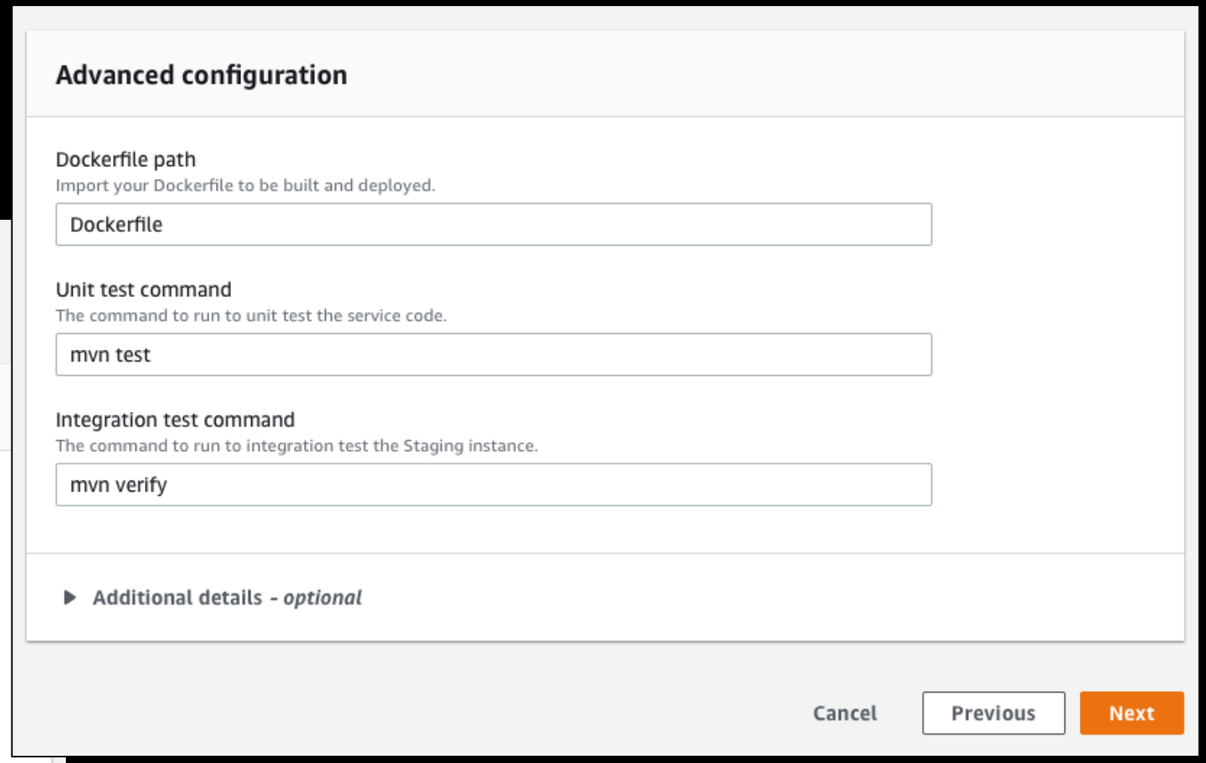
- Write application code
- Select and deploy a service
- Managed deployed services

# Create a service: Select options



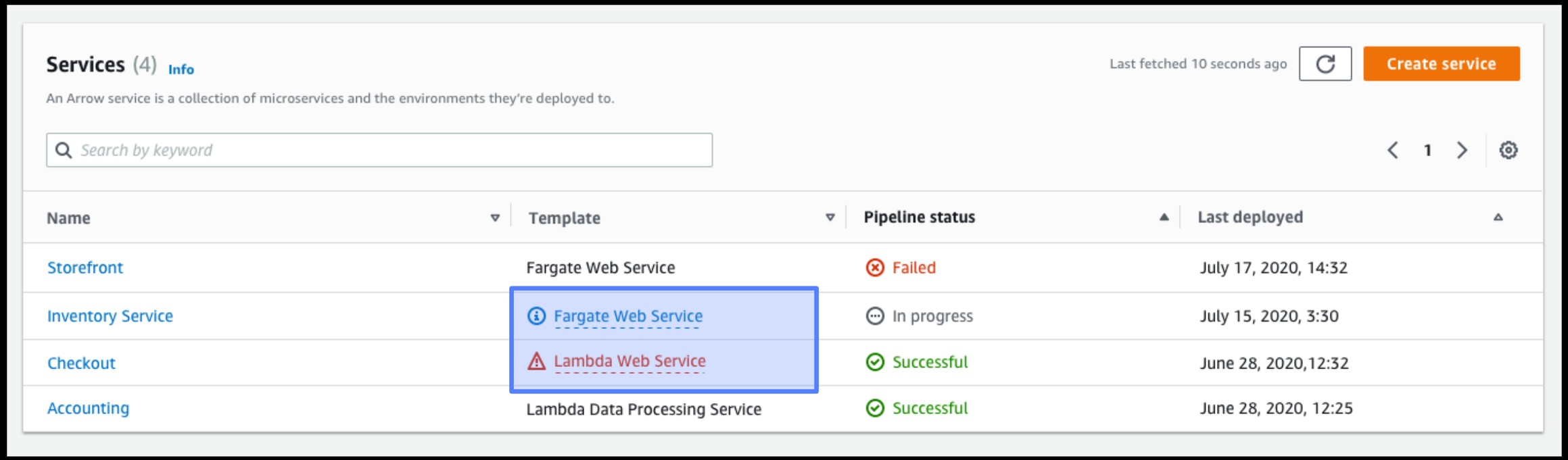
# Create a service: Provide parameters



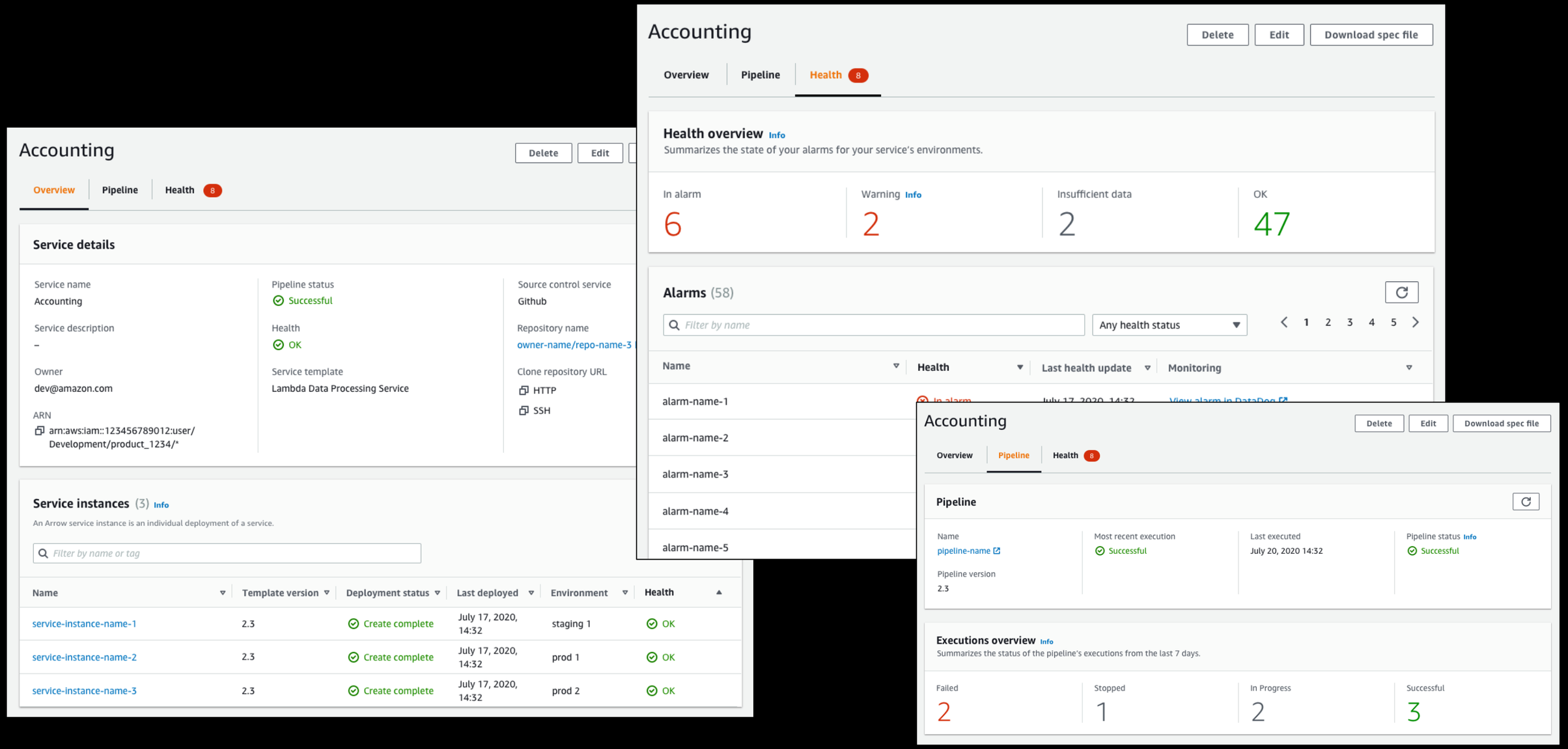




```
DNSRecord:
    Type: AWS:Route53:RecordSet
        Properties:
            Name: {{subdomain}}.example.com
```

# Managing your services



# Detailed view of a service



# Product features

## Public preview

- **Create templates** based on AWS CloudFormation
- Deploy **environments** and **services** based on templates
- Connect **AWS CodePipeline** for CI/CD
- Update templates and **upgrade** deployed environments and services

## Roadmap

- **Health** monitoring
- Support for **third-party tooling**
  - Infrastructure as code, CI/CD, observability
- Automated **tagging** of resources based on template-level rules
- **Multi-account** support
- Template **access control**

# Public preview available now

Go to aws.amazon.com/proton

Visit our public roadmap on GitHub:
github.com/aws/aws-proton-public-roadmap

Check out our sample templates on GitHub:
github.com/aws-samples/aws-proton-sample-templates

# Thank you!



AWS re:Invent

© 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.