
AWS
re:Invent

MGT302

Gaining application-level governance and cost visibility

Karsten Kell

Sr. Manager, Product Management

AWS

AWS re:Invent

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

Agenda

- AWS Service Catalog overview
- Tracking costs for your portfolios and products
- Enabling builder agility
- New feature – AWS Service Catalog AppRegistry
- Using AppRegistry
- Expanding Application Aware Management and Governance



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

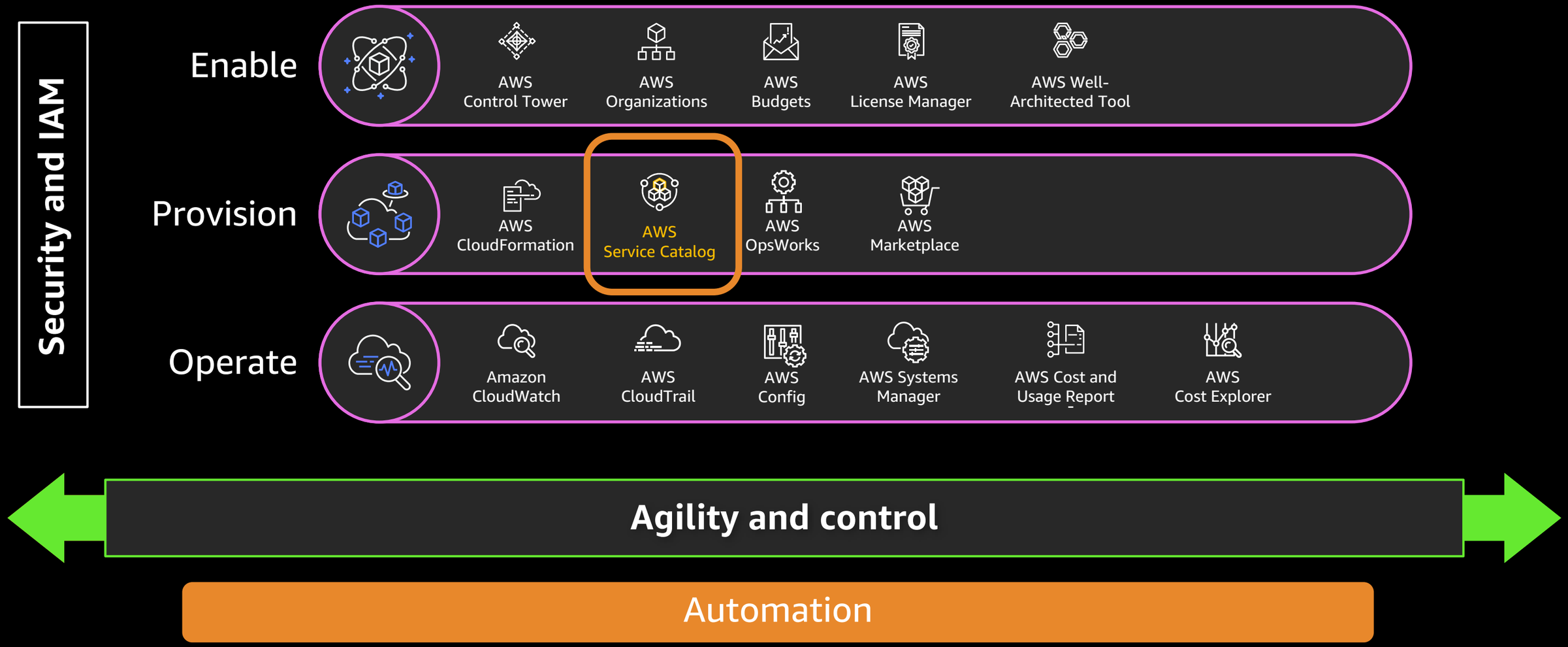
AWS Service Catalog: Balancing agility and governance



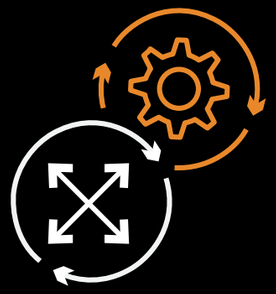
AWS re:Invent

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

AWS Service Catalog is part of AWS Management and Governance Services



More innovation, greater agility, with control



AWS Service Catalog



Get agility **and** control

Agility

Experiment

Be productive

Empower distributed team

Organize, govern, and share products with your teams

Teams self-serve – discover, provision, and operate

Governance

Enable

Provision

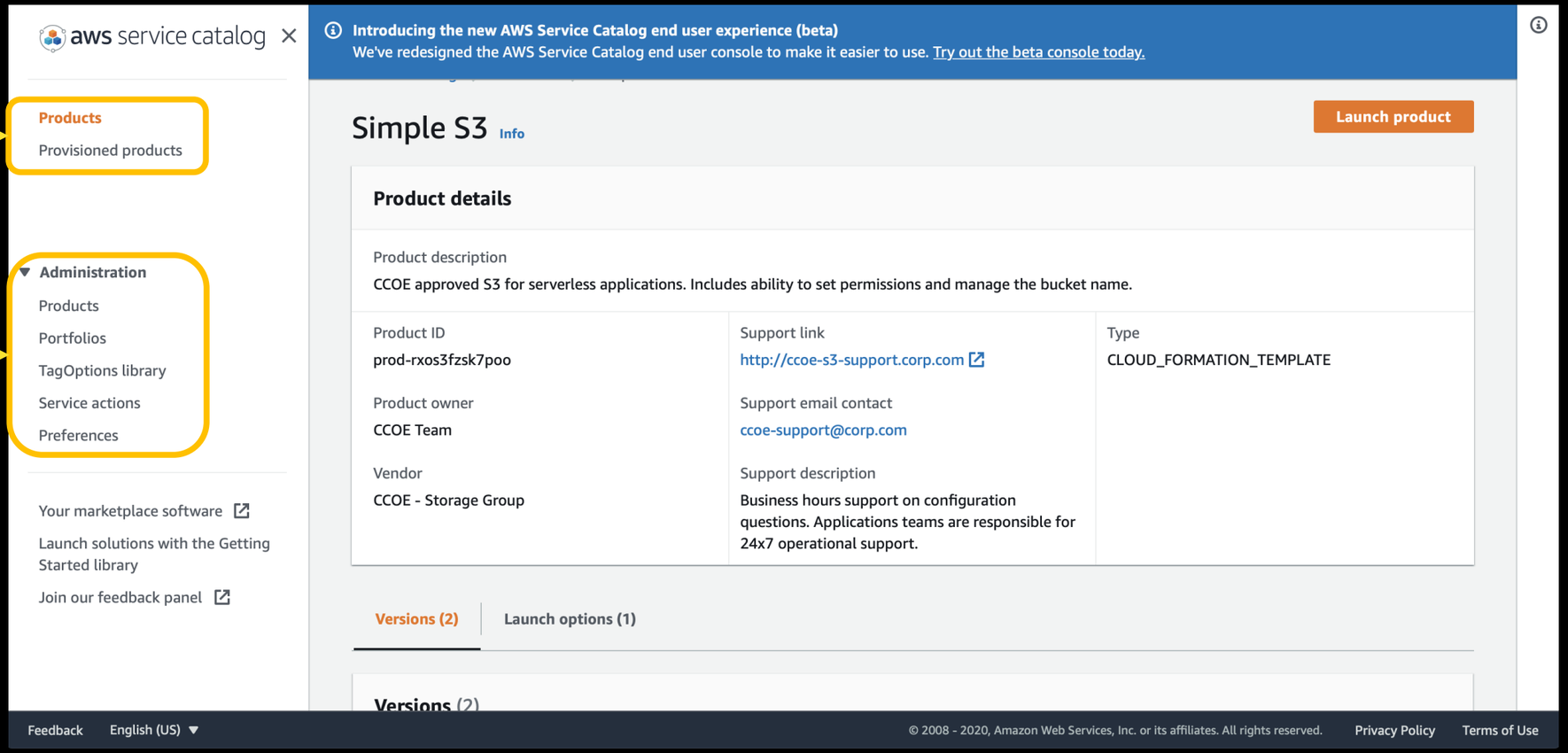
Operate

Organize, govern, share → self-service

1. Cloud Admins establish a catalog of governed best practice products
2. Admins share products with teams
   (acct-to-acct, or via Organizations)
3. Teams self-service discover, provision, and manage resources

End-user →

Administrator →



Self-service products

Anything that you can put in an AWS **CloudFormation** template



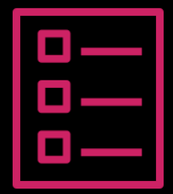
1 Organize products into Portfolios and share with users

2 Add Governance via Service Catalog constraints

[launch, template, StackSet, notification]



Admins can associate TagOptions from their library to products and portfolios



JSON / YAML

AWS CloudFormation template

Product

supports

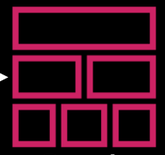
- Service actions
- Versioning
- **TagOptions**

Service Catalog constraints are applied

3 Self-service provisioning

Provisioned product





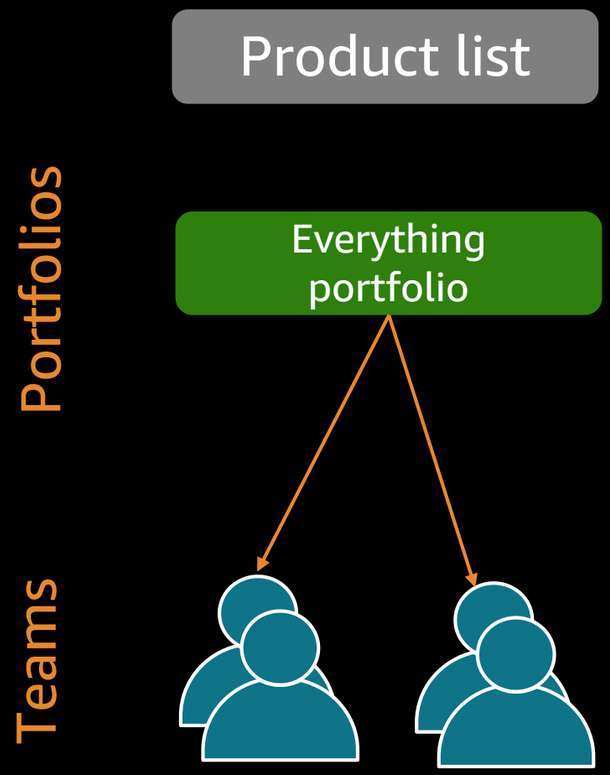
Self-service via

- Console
- API/CLI
- AWS CloudFormation resources

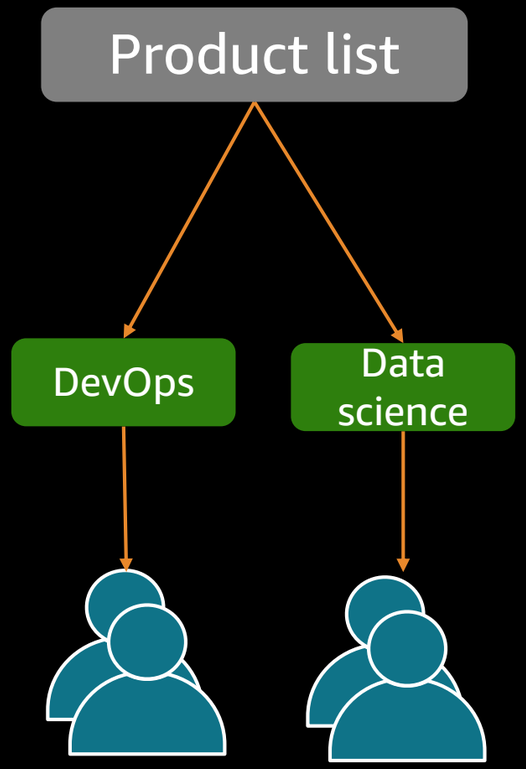
- Properly tagged
- User self-service for operational actions
- User self-service for update, terminate

Customer usage patterns for Service Catalog

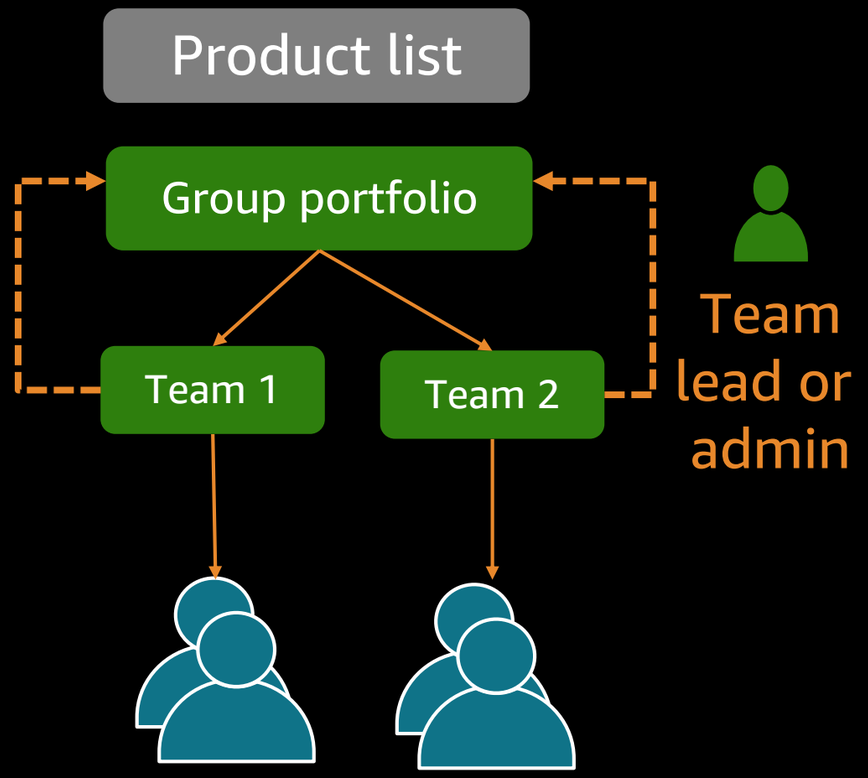
Everyone gets everything



Private catalog shares



With team contributions



Govern your AWS tags with Service Catalog TagOptions and AutoTags

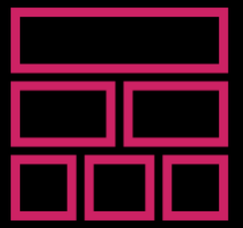




User (Jane)

User selects TagOption values

Provisioned product (Stack)



Portfolio

Product

Provision

Up to 50 user tags

e.g., CostCenter, environment

Associate TagOptions

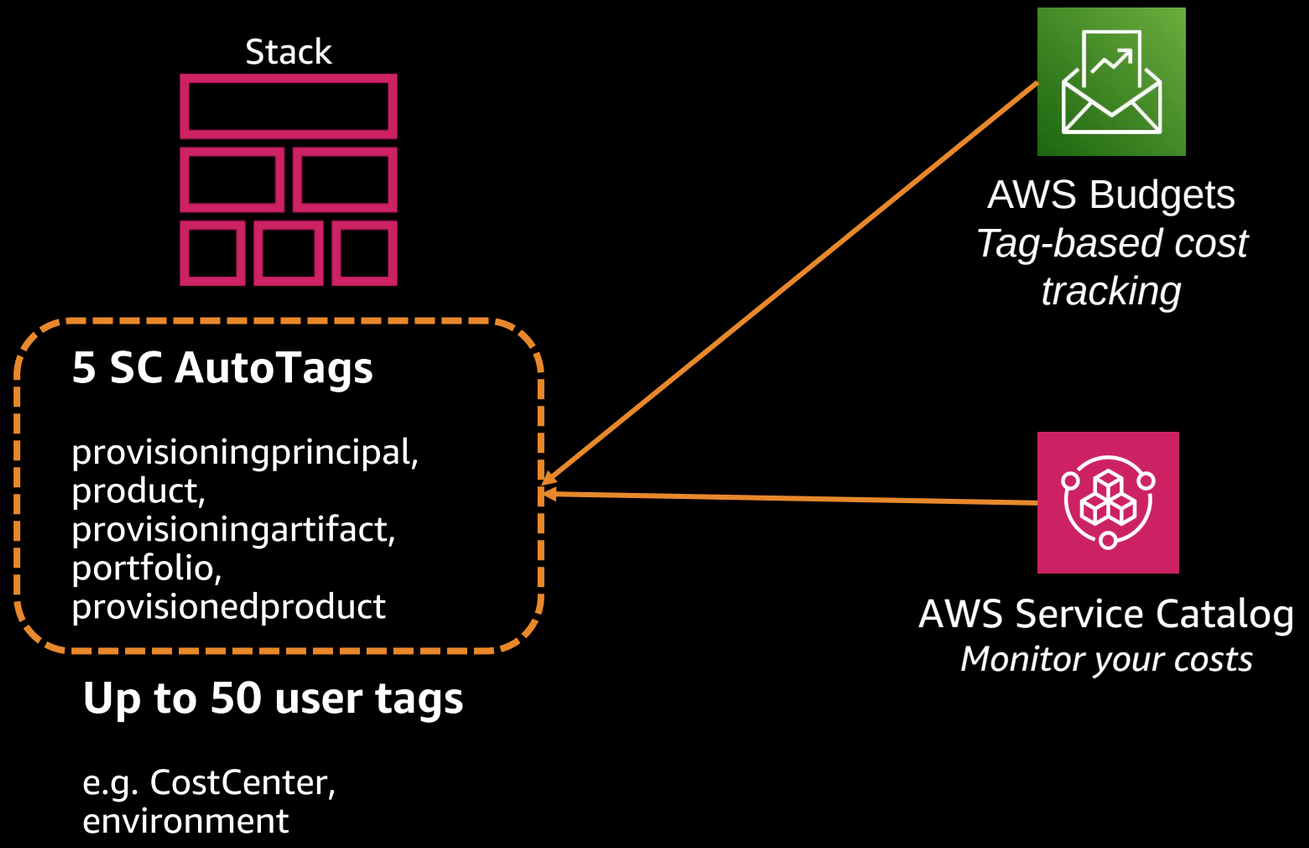
TagOption library

5 Service Catalog AutoTags

provisioningprincipal, product, provisioningartifact, portfolio, provisionedproduct

Monitor product and portfolio costs

AWS Service Catalog operates with AWS Budgets to provide cost monitoring of portfolios and products



1 **Create your budget(s) based on tags, including notification thresholds**

2 **Associate your budget(s) with portfolios and / or products**

and

Monitor your costs

Enabling builder agility



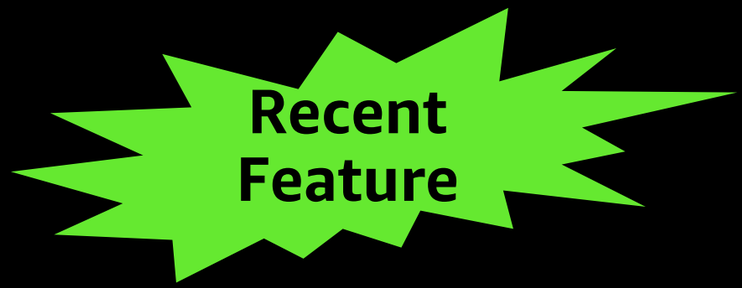
AWS re:Invent

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

Builders want flexibility in how they use Service Catalog products

- **Use** a Service Catalog product in their infrastructure definition (e.g., provision a Service Catalog product from within an AWS CloudFormation template)

- **Combine** multiple Service Catalog products into a more complex product (e.g., an N-tier stack using admin-supplied VPC, EC2, and RDS products)



Builders use compliant, best practice products in their application infrastructure

1. Admins distribute approved products to builders

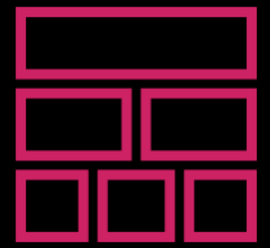


2. Builders use products in their application infrastructure code

```
AWS CloudFormation template

Type:
AWS::ServiceCatalog::CloudFormation
ProvisionedProduct
Properties:
...
```

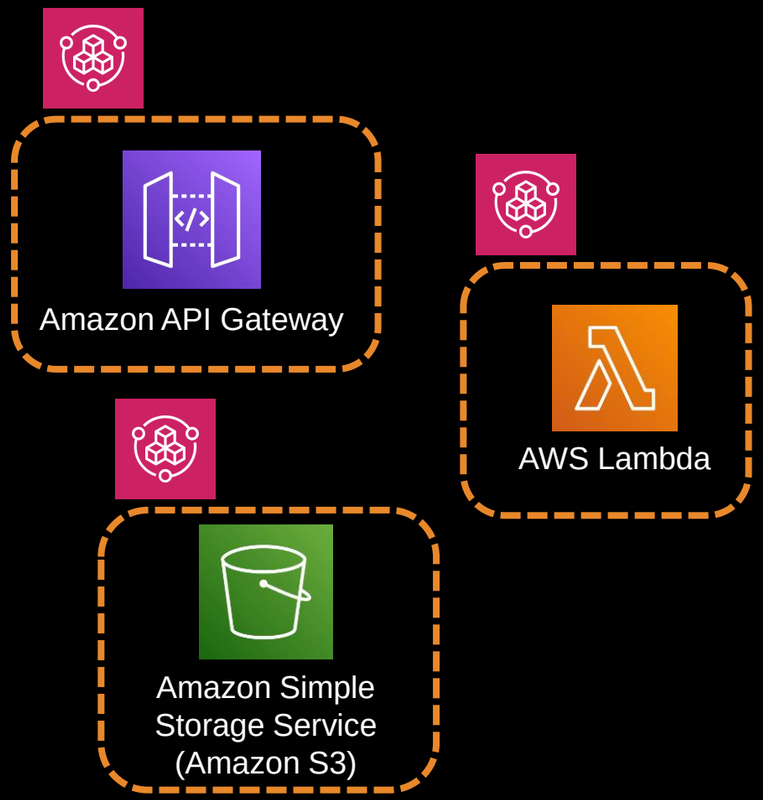
3. Builders provision applications through AWS CloudFormation



Stack

Builders combine compliant, best practice products

1 Admins distribute approved products to builders

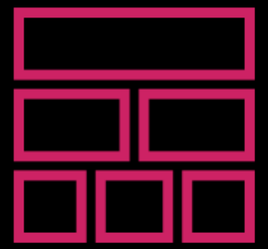


2 Builders combine Service Catalog products in a template

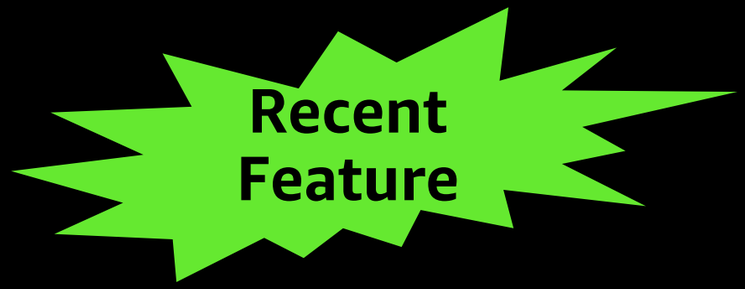
```
AWS CloudFormation template

Type:
AWS::ServiceCatalog::CloudFormation
ProvisionedProduct
Properties:
…
Value: !GetAtt  …
```

3 Builders provision applications through AWS CloudFormation



Stack



Combine two SC products: Serverless stack

```
DeAWSTemplateFormatVersion: '2010-09-09'
scription: Serverless
Stack Parameters:
  inputDomainName:
  Type: String

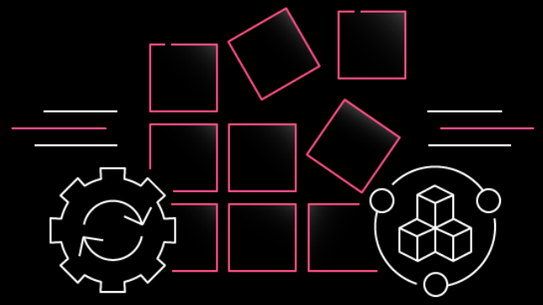
Resources:
 SimpleLambda:
   Type: AWS::ServiceCatalog::CloudFormationProvisionedProduct
   Properties:
    ProductName: Basic Lambda Provisioning
    ArtifactName: '1.0'

 SimpleApiGateway:
   Type: AWS::ServiceCatalog::CloudFormationProvisionedProduct
   Properties:
    ProductName: API Gateway
    ProvisioningArtifactName: '1.1'
    ProvisioningParameters:
   -
     Key: myLambdaArn
     Value: !GetAtt [SimpleLambda, Outputs.myLambdaArn]
   -
     Key: myDomainName
     Value: !Ref inputDomainName
```

Define a Lambda

Define your API Gateway

Connect your Lambda to your API Gateway



New feature: AWS Service Catalog AppRegistry

AWS re:Invent

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

The need for an application-level abstraction



- I need a way to collect all of my application instances (workloads) across my dev, test, and prod environments and across my teams
- I need to associate data from different departments, internal systems, ISV tools, and teams to provide a holistic view

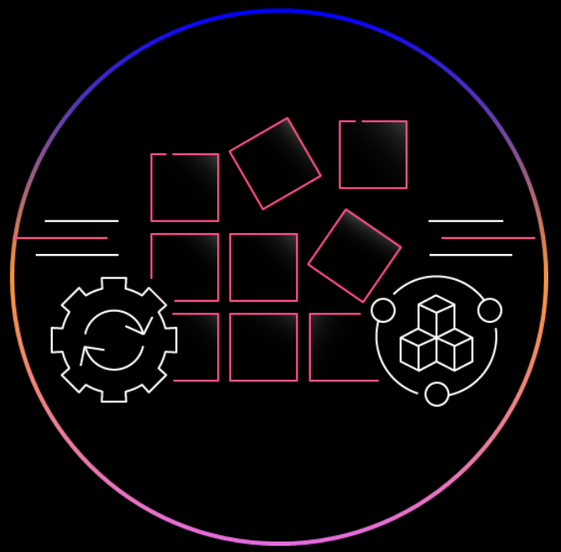
- How do I know if this EC2 instance is part of a production workload?
- I use AWS Systems Manager to perform operations on my application resources
- If I perform an operation (e.g., patch), which workloads are impacted?

INTRODUCING

AVAILABLE TODAY

AWS Service Catalog AppRegistry

DEFINE AND MANAGE APPLICATIONS AND METADATA



Create application definitions
Including resource collections and metadata from AWS services and ISV partners

Maintain a single source of truth
By integrating AppRegistry with your application development processes

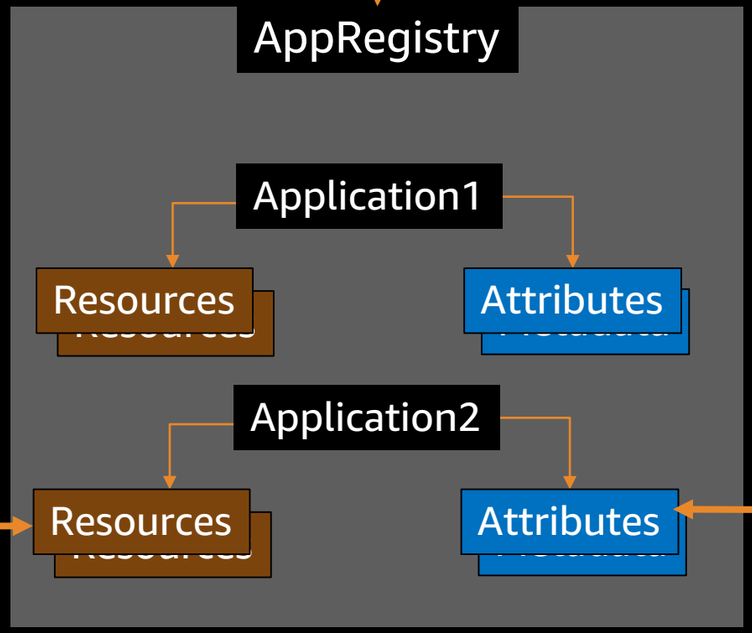
Get application context for your resources
Know what application your resource belongs to and vice versa

AppRegistry: Associate your stacks and extensible metadata to an application object

Application
A new application object to which you associate your application resources (AWS CloudFormation Stacks)



AWS Service Catalog
Define your applications



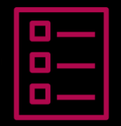
Resources
Application resource collections (e.g., an AWS CloudFormation Stack)



Stack(s)

Associate

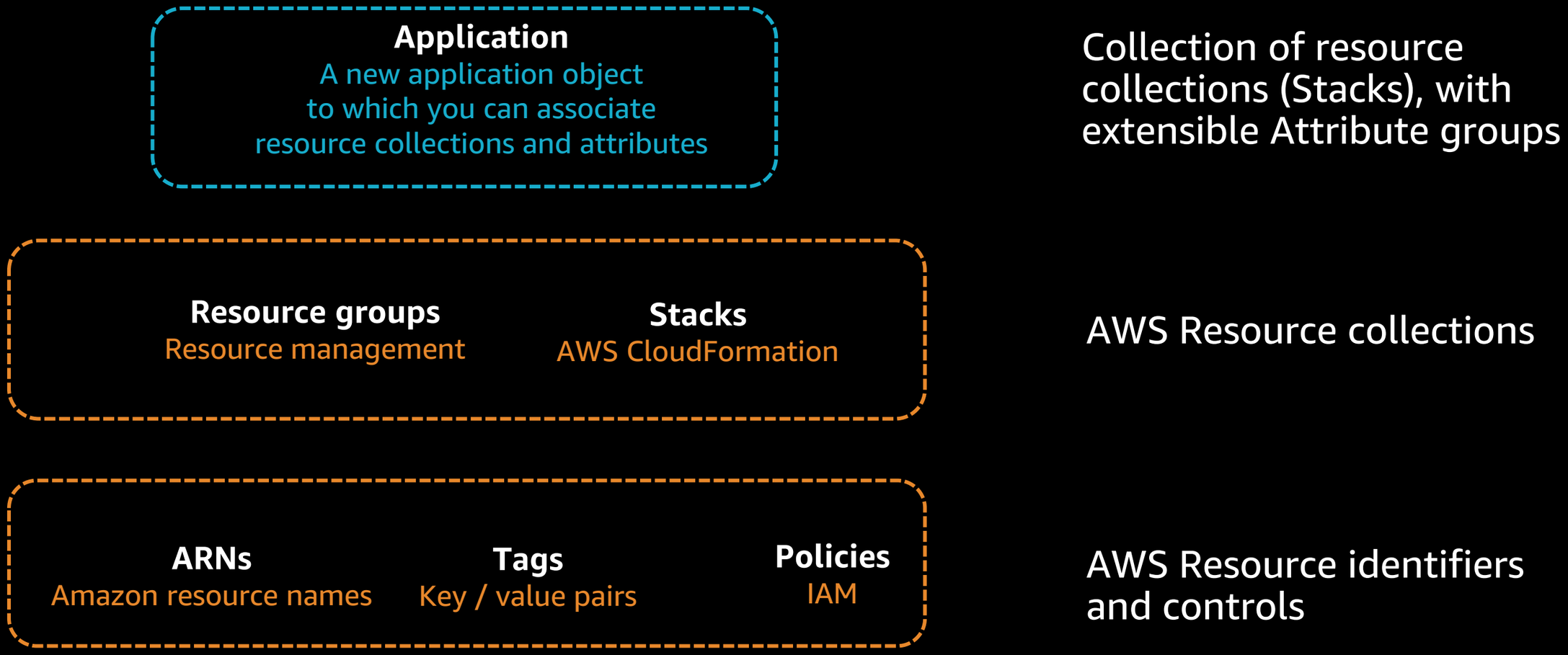
Attribute groups
JSON file(s) containing application metadata



JSON file(s)

Associate

Built on top of familiar AWS constructs



Collection of resource collections (Stacks), with extensible Attribute groups

AWS Resource collections

AWS Resource identifiers and controls

Defining your application

1. Create your application and associate resources (Stacks)

```
> aws servicecatalog-appregistry create-application
> aws servicecatalog-appregistry associate-resource
```

2. Create your attributes (JSON files) and association them to your application

```
> aws servicecatalog-appregistry create-attribute-group
> aws servicecatalog-appregistry associate-attribute-group
```

Understanding your application

1. **LIST your applications, attribute groups**

```
> aws servicecatalog-appregistry list-applications
> aws servicecatalog-appregistry list-attribute-groups
```

2. **GET application details (associated Stacks and attribute groups)**

```
> aws servicecatalog-appregistry get-application
> …  list-associated-resources
> …  list-associated-attribute-groups
> aws servicecatalog-appregistry get-attribute-group
```

3. **Update, disassociate, delete**

```
> aws servicecatalog-appregistry update-attribute-group
> … disassociate-attribute-group
> … disassociate-resource
> aws servicecatalog-appregistry delete-attribute-group
```

Applications provide context for resources

List my applications

Get application detail (list of Stacks)

Get attribute detail(s)

Drill into Stack / resource details

Applications
Logically grouped resource collections with rich metadata

Stacks
AWS CloudFormation

Tags
AWS System tags and user tags

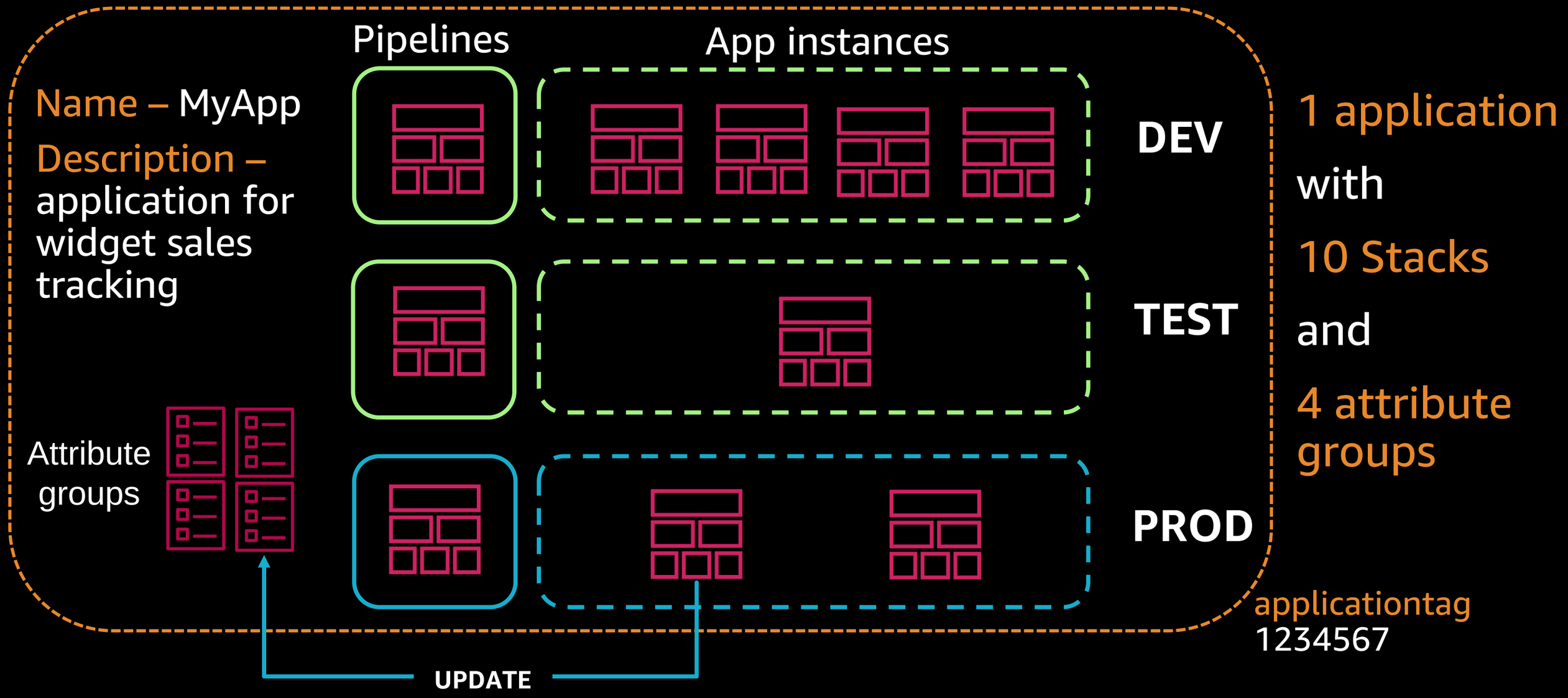
Make decisions based on rich metadata

Get application detail (list of Stacks)

Get attribute detail(s)

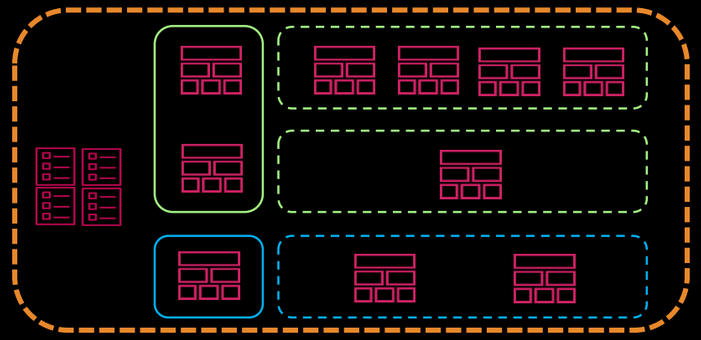
Examine resource tags (applicationArn)

A sample application definition



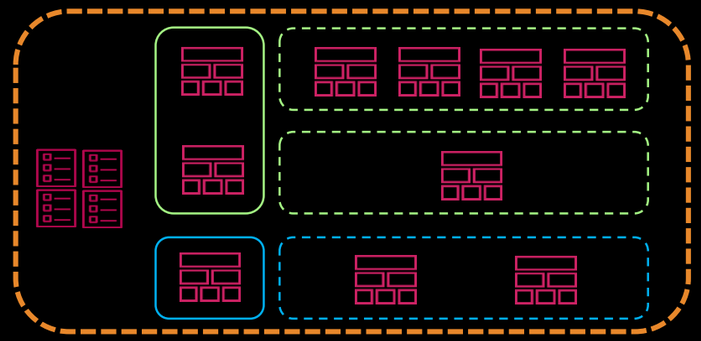
Track costs for your applications

MyApp



applicationtag
1234567

MyOtherApp



applicationtag
7654321

ISV software
cost monitoring, cost analysis, cost allocation



AWS Budgets
Tag-based cost tracking

Customer benefits driven by AWS partners

“We allow customers to easily manage AWS Service Catalog portfolios and provide a method to easily push out roles to help limit required user permission while providing autonomy to your teams. With AWS Service Catalog Application Registry, we can now help segment complex application workloads and track changes in resources across AWS accounts. This enables customers to better organize their information and visualize where their applications live so it's less of a worry when making broad changes across the organization.”

Joseph Spurrier, CTO
cloudtamer.io

“AWS Service Catalog AppRegistry will help Cutover simplify the migration of applications to AWS and the orchestration of change once there, by making it easy to have a central registry of all applications. AppRegistry allows us to automate the registration of application resources and metadata as they change to provide application context as part of an application deployment pipeline at cloud scale.”

Ky Nichol, CEO
Cutover



“Our joint customers need to understand the complexity of their systems and applications, and manage them at cloud scale. That’s why New Relic is excited to work with AWS on their launch of AWS Service Catalog AppRegistry and work together on solving our customers’ problems.”

Bill Staples, Chief Product Officer
NewRelic



“AppDynamics and AWS helps our customers by ensuring application performance through their migrations and adoption to the cloud. AppDynamics is excited to continue partnering with AWS on AppRegistry.”

Vipul Shah, Chief Product Officer
AppDynamics

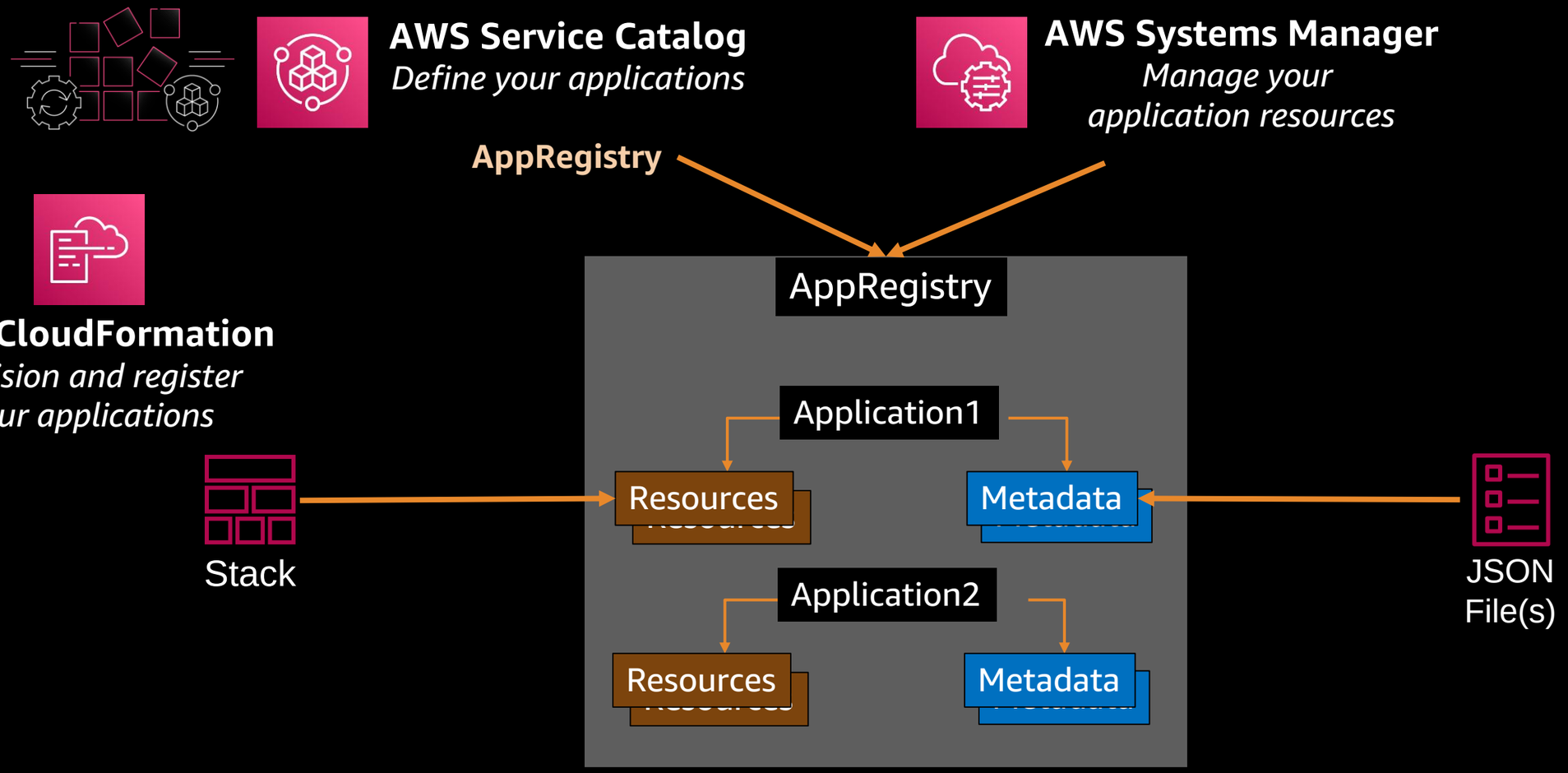
Application aware management and governance



AWS re:Invent

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

Application aware management and governance



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.