



Prototyping and Cloud Engineering



Financial Services Industry

OPA on AWS

Platform Engineering



Orchestrate Platform and Applications

FSI PACE

Platform Engineering Team

fsi-pace-pe@amazon.com

Architecting your customers needs

- 1 Tech stack and applications requirements
- 2 Required environments and operating methodology
- 3 Apply security guardrails and organization standards
- 4 Implement observability and governance control
- 5 Delegate access for teams to own their apps, infrastructure and environments

Example

- 1 HR Team builds Java SpringBoot application with Amazon RDS
- 2 Team wants to run their applications on EKS and wish to use three separate environments: dev, test, prod all in eu-west-1 region
- 3 Organizational standards require Snyk and Splunk to be integrated into all applications and pipelines
- 4 Implement AWS CloudTrail and Amazon Managed Grafana and Prometheus
- 5 Grant HR team AD group access to create their apps, infrastructure and environments

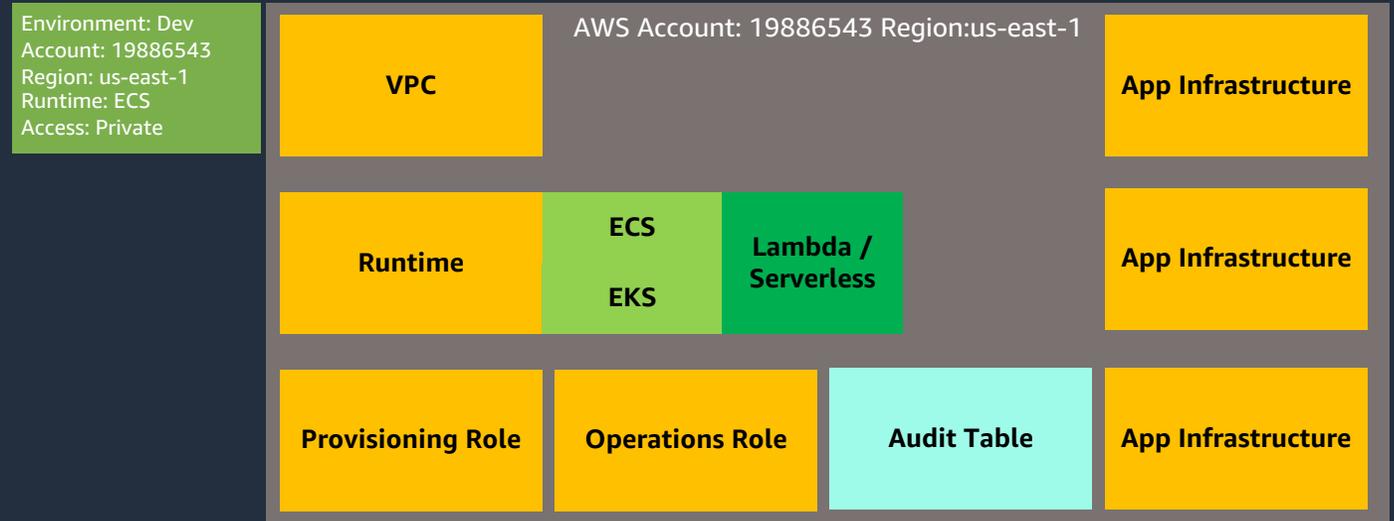


Creating Providers



AWS Environment Provider

1. Contains underline network, runtime environment, operating and provisioning roles, audit table and applications infrastructure
2. Exists within a specific account and region
3. Mutually exclusive – within an account and region multiple providers can be created
4. Isolated from other providers / accounts
5. Supports prefix for organization segmentation



Demo

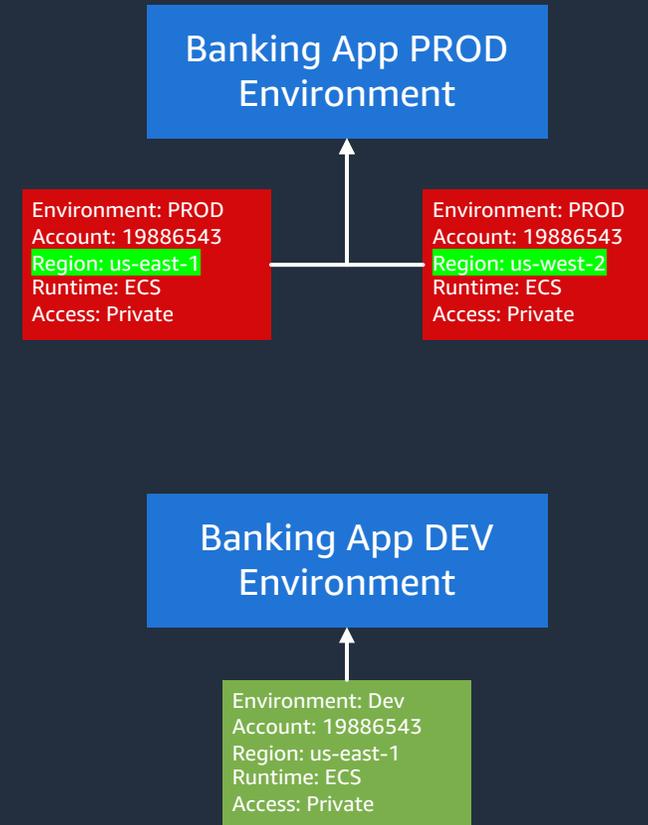


Creating Environments



AWS Environment

1. Abstracted entity
 1. Single/multi account
 2. Single/multi region
 3. Category – dev, test, stage, prod etc.
 4. Classification – private, internal, public
 5. Hierarchy – where does it position in the hierarchy of other environments (low– dev, high – prod)
2. Maintain 1:N relationship with AWS environment provider
3. Integrated with pipeline definition for require approval deployments
4. Customizable and extendable



Demo



Creating Apps and Resources



Creating Templates

Apps

- ✓ **Single codebase** over multiple environments
- ✓ Deployment for **IAC** is executed and controlled over pipeline
- ✓ Deployment for **application code** is built and executed over a pipeline
- ✓ Operating **multiple environments** through a single interface
- ✓ Access logs, variables, start/stop, audit info, bounded resources **through one page**
- ✓ Deployment to other environments is **automated** through pipeline
- ✓ Support for **different flavors of apps** and technologies – TF, CDK, Java, Python, Node.js

Shared Resources

- ✓ **Single codebase** per environment
- ✓ Deployment for **IAC** is executed and controlled over pipeline
- ✓ Access resource info through **resource page**
- ✓ **Bind resources** to application with predefined permissions template
- ✓ Support for different flavors of IAC Including – **Terraform and CDK**

Demo





Thank you!



Orchestrate Platform and Applications

fsi-pace-pe@amazon.com