



What are Containers?

Containers are lightweight, standalone, executable software packages that include every libraries, and settings. They provide:

- **Consistency:** Run the same way regardless of the infrastructure
- **Isolation:** Applications run in isolated environments without interfering with each other
- **Efficiency:** More lightweight than virtual machines, sharing the host OS kernel
- **Portability:** Run anywhere - development, testing, and production environments
- **Scalability:** Easy to scale horizontally by deploying more container instances



What are Containers?

Containers are lightweight, standalone, executable software packages that include every libraries, and settings. They provide:

- **Consistency:** Run the same way regardless of the infrastructure
- **Isolation:** Applications run in isolated environments without interfering with each other
- **Efficiency:** More lightweight than virtual machines, sharing the host OS kernel
- **Portability:** Run anywhere - development, testing, and production environments
- **Scalability:** Easy to scale horizontally by deploying more container instances

Agenda

- ① Intro to Containers
- ② Building a docker file
- ③ ECR - Basics
- ④ Pushing an Image to ECR
- ⑤ Introduction to App Runner
- ⑥ Running an application on App Runner
- ⑦ Basics of EC2

Container Registries



What is a Container Registry?

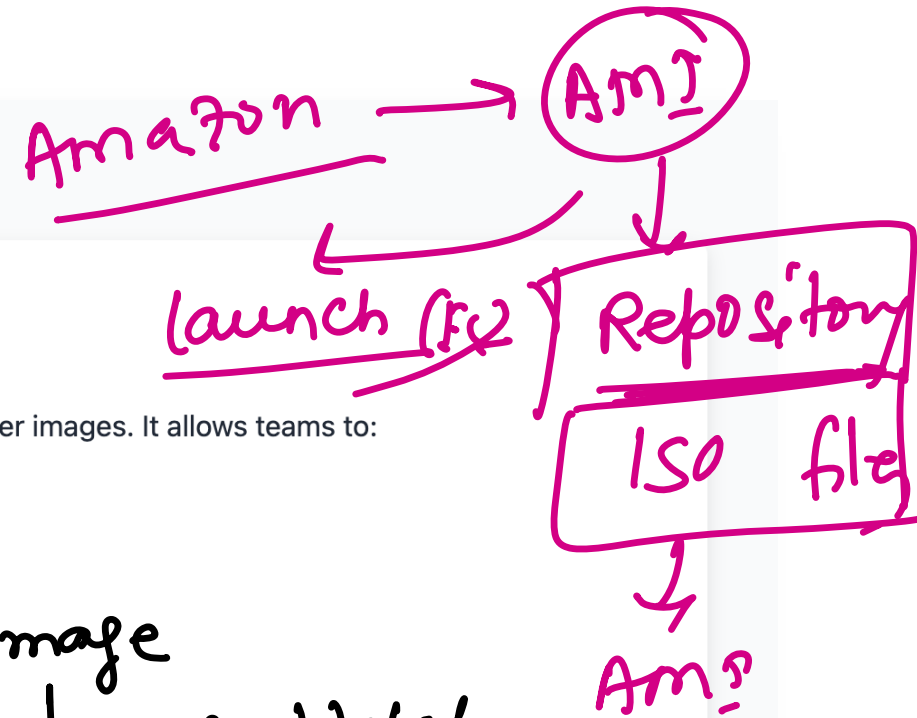
A container registry is a repository or collection of repositories used for storing and distributing container images. It allows teams to:

- Store container images in a centralized location
- Version and tag images for different environments
- Share images across teams and deployment environments
- Implement access controls and security scanning
- Automate CI/CD pipelines with image builds and deployments

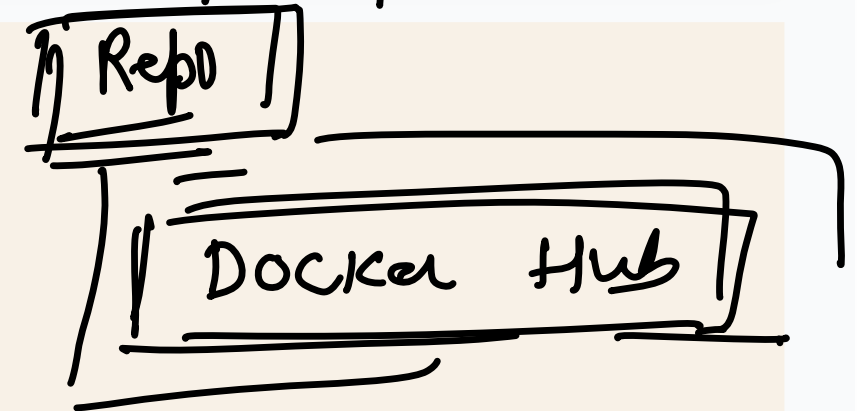
Why Do We Need Container Registries?

Container registries solve several critical challenges in modern application development:

- **Distribution:** Easily share container images across teams and environments
- **Version Control:** Track changes and maintain different versions of images
- **Security:** Scan images for vulnerabilities before deployment
- **Automation:** Enable CI/CD pipelines with automated builds and deployments
- **Scalability:** Handle high-volume pulls during large-scale deployments



Docker Image
↓ To be played



Container Basics

Containers vs VMs

Docker Workflow

Image In Running state is Container

Containers provide a consistent environment for applications to run, regardless of the host system.

Application Code

Your application source code and dependencies

Containerization

Package with Docker into a container image

Distribution

Push to a container registry

Deployment

Run anywhere with Docker runtime

① AMI
↓
Launch VM
↓
Configure App.
↓
Task = AMIs with code
↓
put in ASH

Docker file

Steps: Install python
Install libraries
App code
git — start

VM

Running inside VM

Run.

FCR

Container Basics

Containers vs VMs

Docker Workflow

Virtual Machines

- ✓ Full OS in each VM
- ✓ Hypervisor required
- ✓ Slower startup time
- ✓ More resource intensive
- ✓ Complete isolation

Containers

- ✓ Share host OS kernel
- ✓ No hypervisor needed
- ✓ Fast startup time
- ✓ Lightweight
- ✓ Process-level isolation

commend

App Runner

ECS

EKS

Fargate

AWS App Runner

Best for: Simple web applications and APIs with minimal configuration

- ✓ Fully managed infrastructure
- ✓ Automatic scaling
- ✓ Built-in CI/CD
- ✓ HTTPS by default
- ✓ Private networking with VPC

Source Code or Container

Your application code or container image



App Runner Service

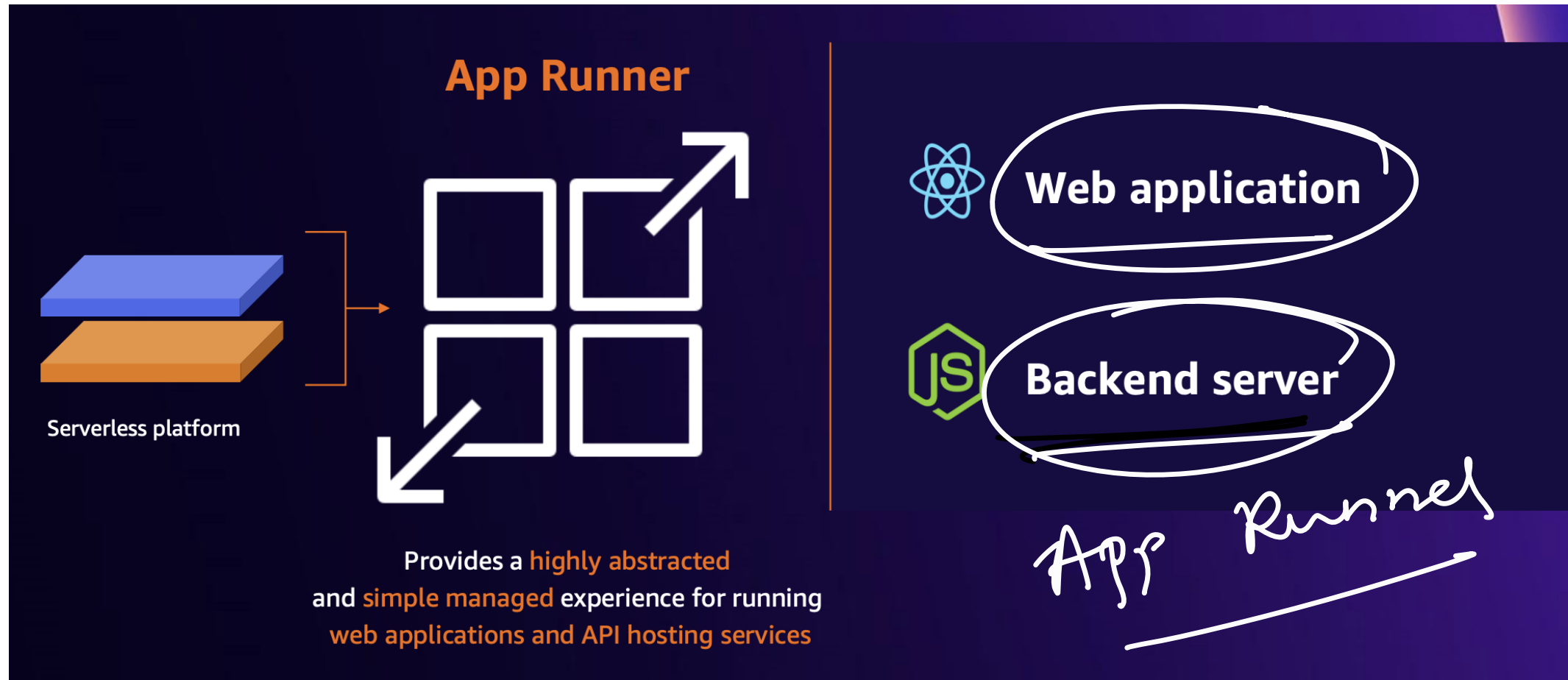
Builds, deploys, and manages your application



Production Deployment

Automatic scaling, load balancing, and HTTPS

Fully managed service for web application





Move to cloud

Fully managed application service,
built-in AZ resiliency, integrated
load balancer

App Runner



Build and deploy

Integrated source code providers and
container registries, automatic builds,
and deployments



Networking

Public and private ingress,
VPC Egress. Integrated with
AWS Secrets Manager, and
AWS Systems Manager



Security

Integrated AWS WAF

Layer 7



Auto scaling

Auto scaling on concurrent

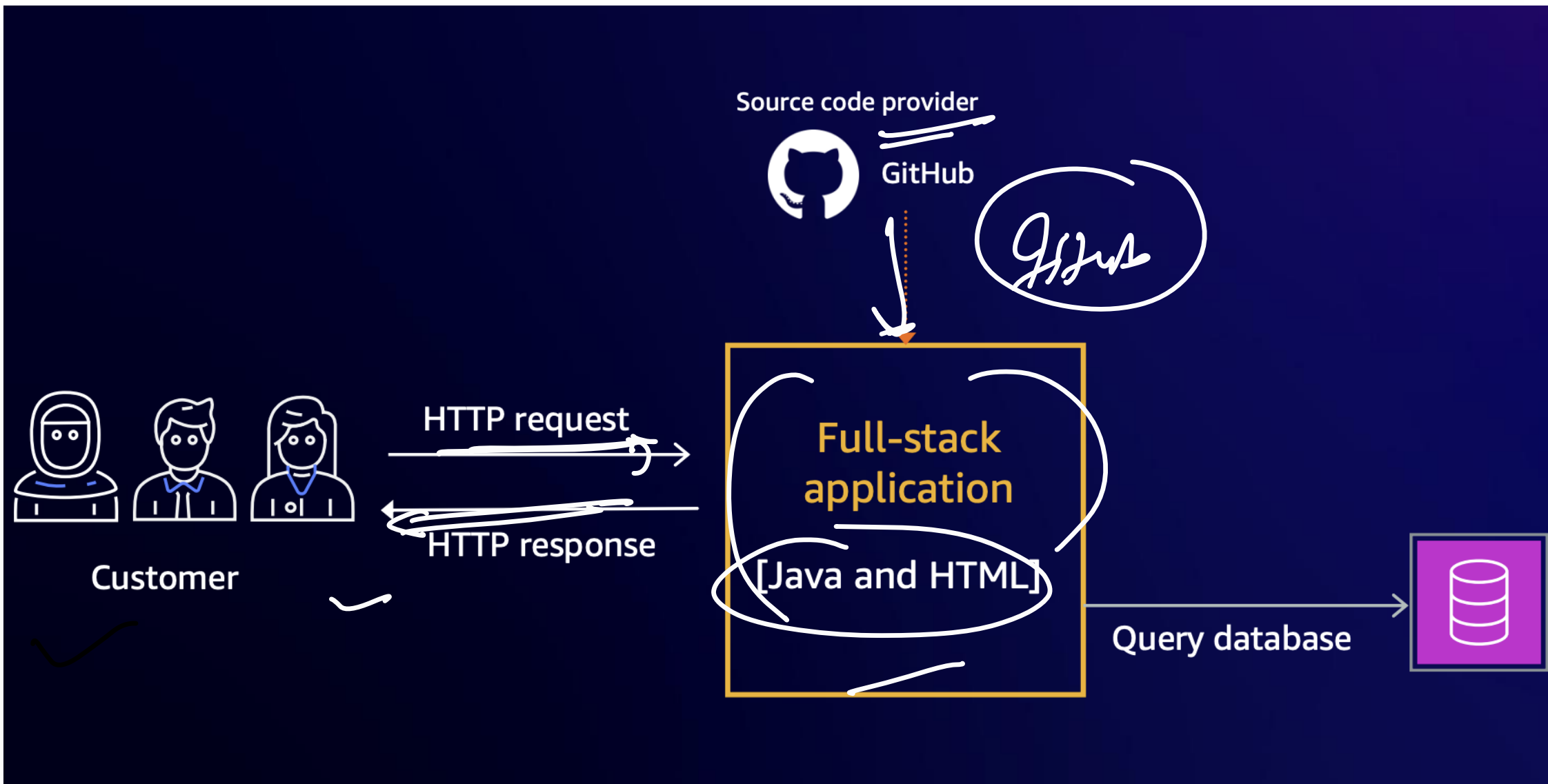


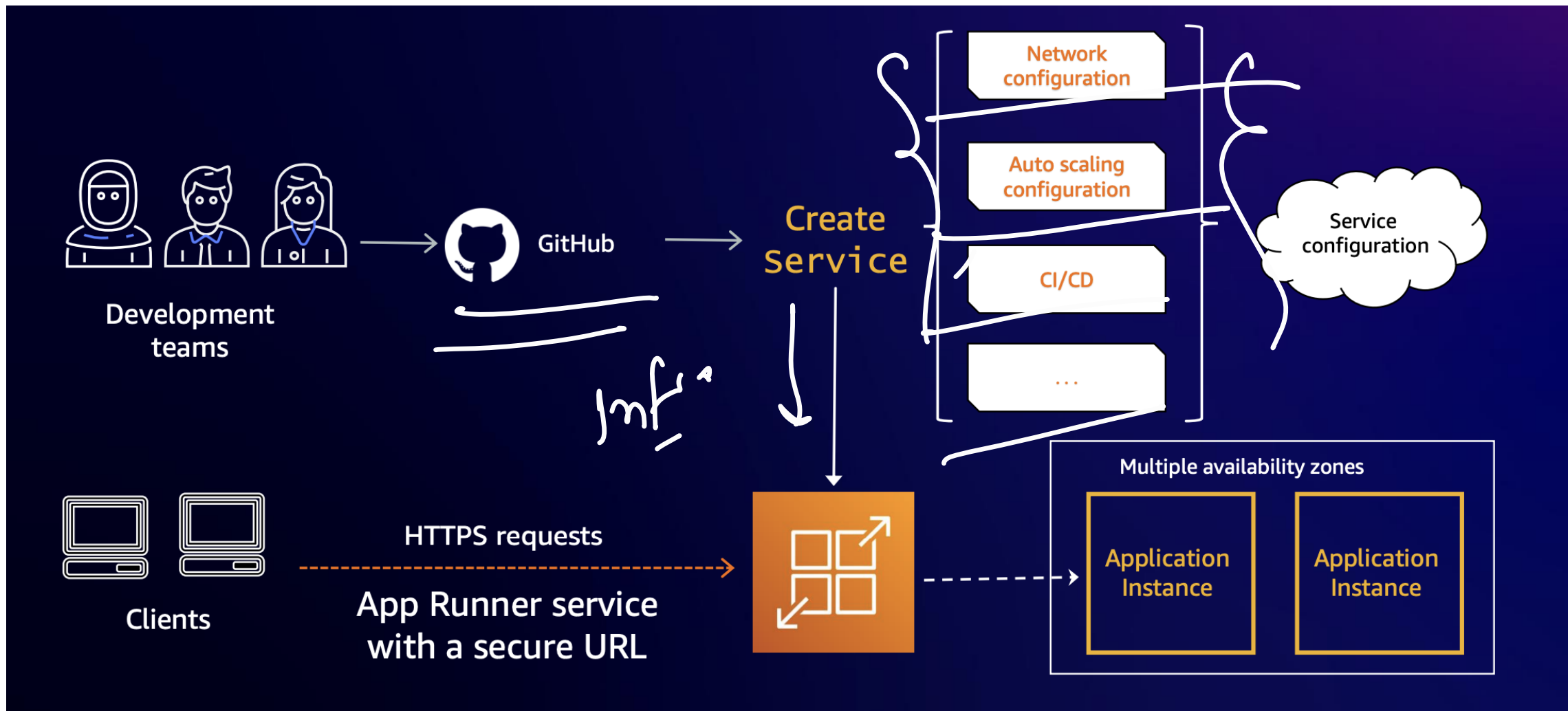
Observability

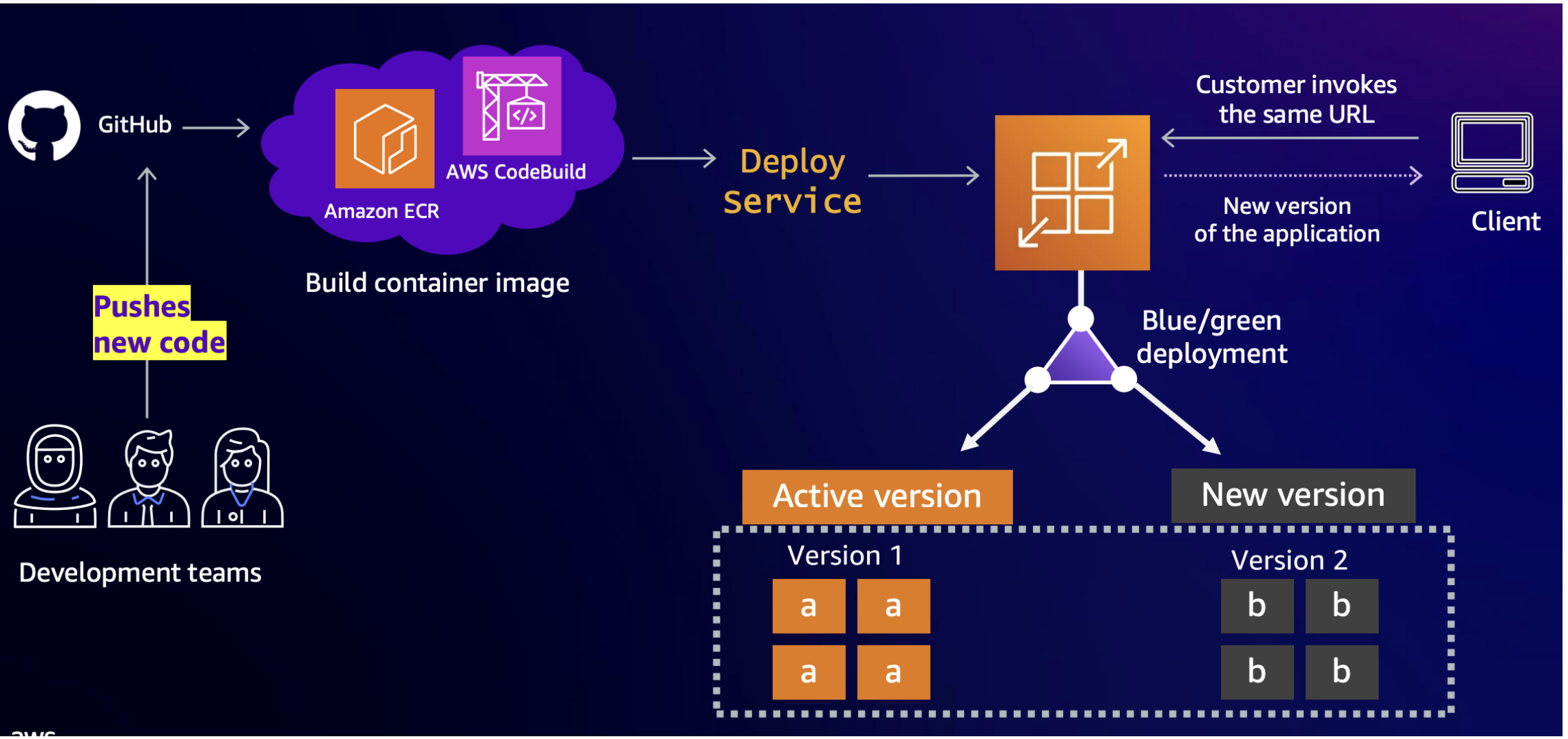
Integrated Amazon Cloud
AWS X-Ray

- Demo <https://github.com/hariohmprasath/demo-app>











Requests (100) →
10 queued (cold start)



App Runner
service URL

30

Application
Instance

30

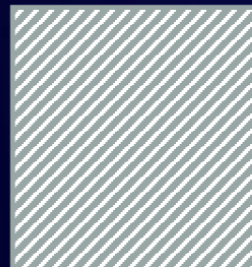
Application
Instance

30

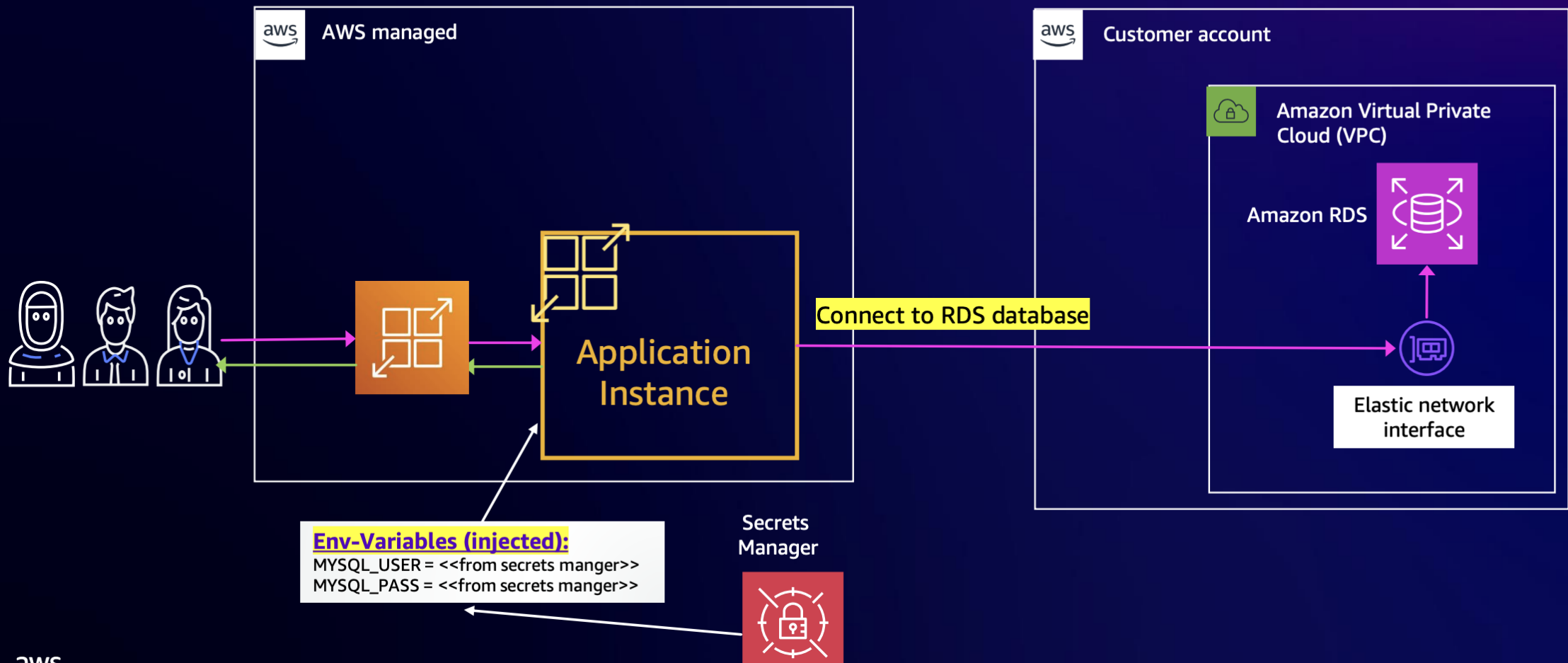
Application
Instance

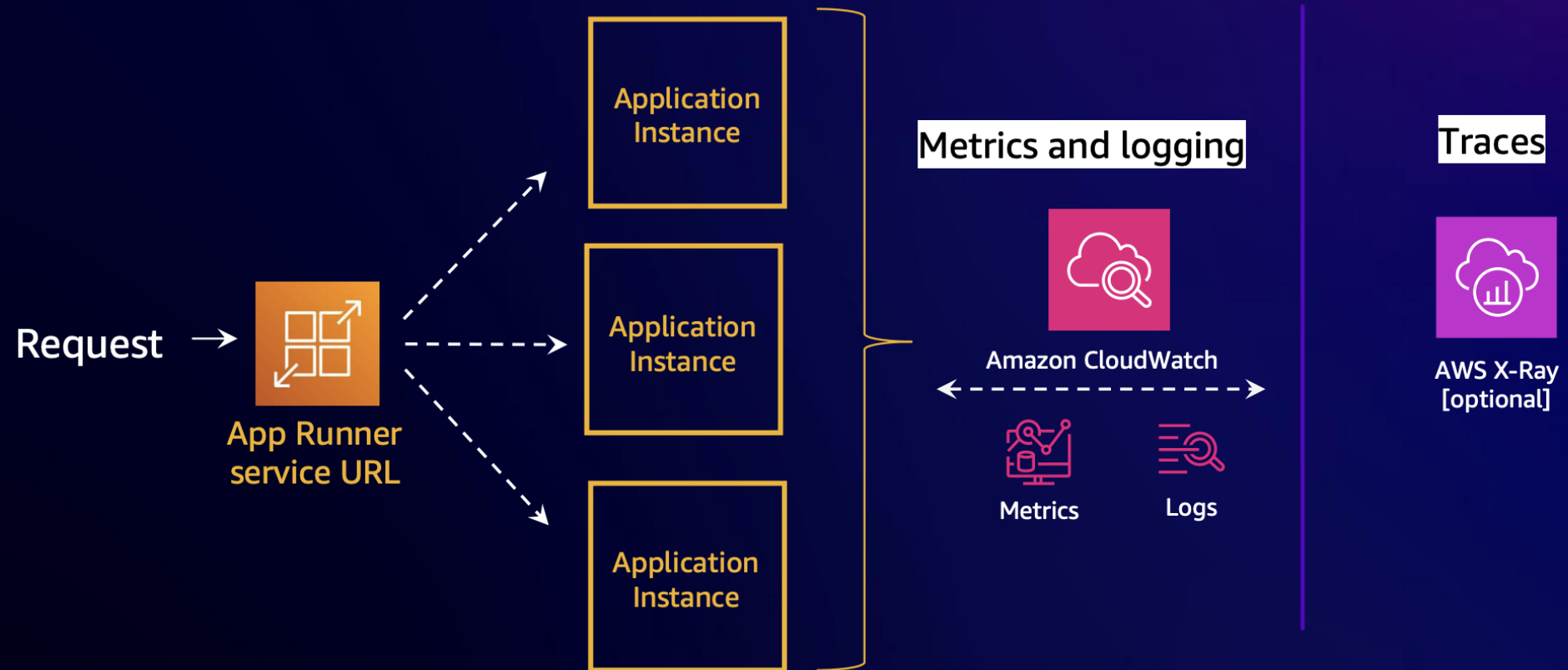
Scalable configuration

concurrency = 30; min=3; max=5



Launching new instance





App Runner provides

- Simple **abstracted experience** for HTTP(S) requests/reply-based web applications
- Faster development cycles and enables existing applications to be production ready in **minutes**
- **Continuous build and deploy capabilities** integrated with source code repositories
- Integrated **auto scaling and observability**
- Built-in **AZ resiliency and fault tolerance**
- Protects your applications with **AWS WAF**, integrates seamlessly with **Secrets Manager and Systems Manager** for accessing sensitive data