Managing Secrets in AWS

Justin Menga
FULL STACK TECHNOLOGIST
@jmenga pseudo.co.de

Introduction

Managing Secrets in AWS

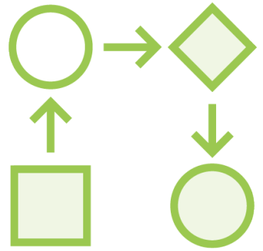
- Secrets challenges
  - Secure storage of secrets
  - Consuming secrets in Docker
- So what's the solution?
  - AWS EC2 Systems Manager (SSM)
  - AWS Key Management Service (KMS)
  - CloudFormation custom resource
  - Docker container support
  - CloudFormation resource support

Introducing the EC2 Systems Manager

EC2 Systems Manager Features

Run Command

Automation Tasks



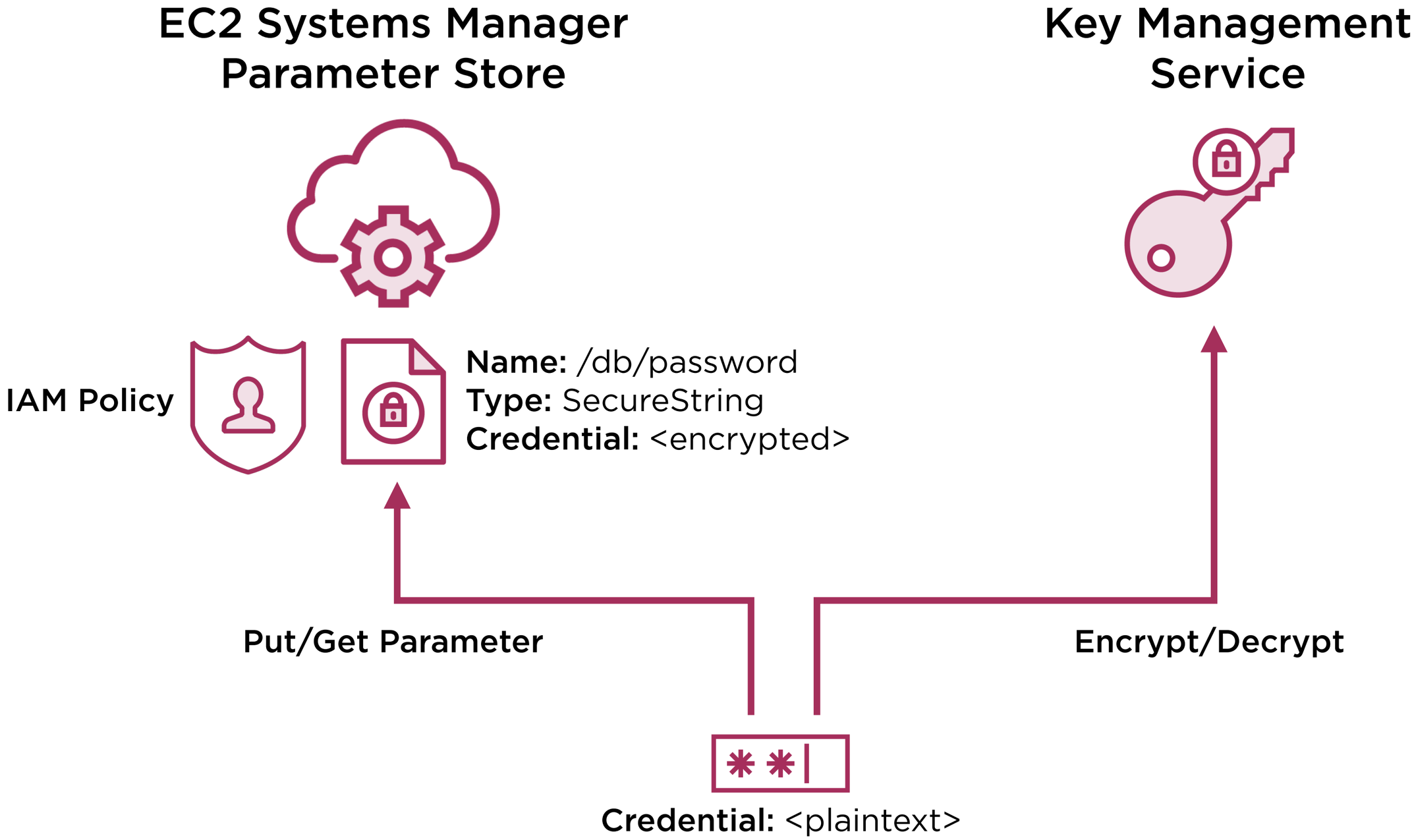
State Manager

Inventory Management

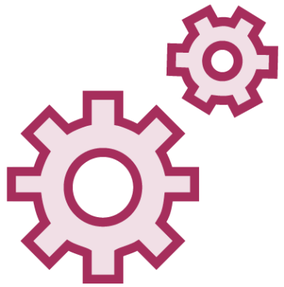
Patch Manager

Parameter Store

See **https://aws.amazon.com/ec2/systems-manager/** for more details


EC2 Systems Manager
Parameter Store
Key Management
Service
IAM Policy
Name: /db/password
Type: SecureString
Credential: <encrypted>
Put/Get Parameter
Encrypt/Decrypt
Credential: <plaintext>

Secrets Management Solution Overview



SSM Secrets Provisioner

Resource Properties
Name: /${AWS::StackName}/db/password
Key: JDBC_PASSWORD
Value: <value or random if omitted>
KmsKeyId: <kms key-id>

SSM Secret Custom Resource(s)

Resource Properties

`Name: STRING`

- **Must be unique**
  e.g. /${AWS::StackName}/db/secret

`Key: STRING`

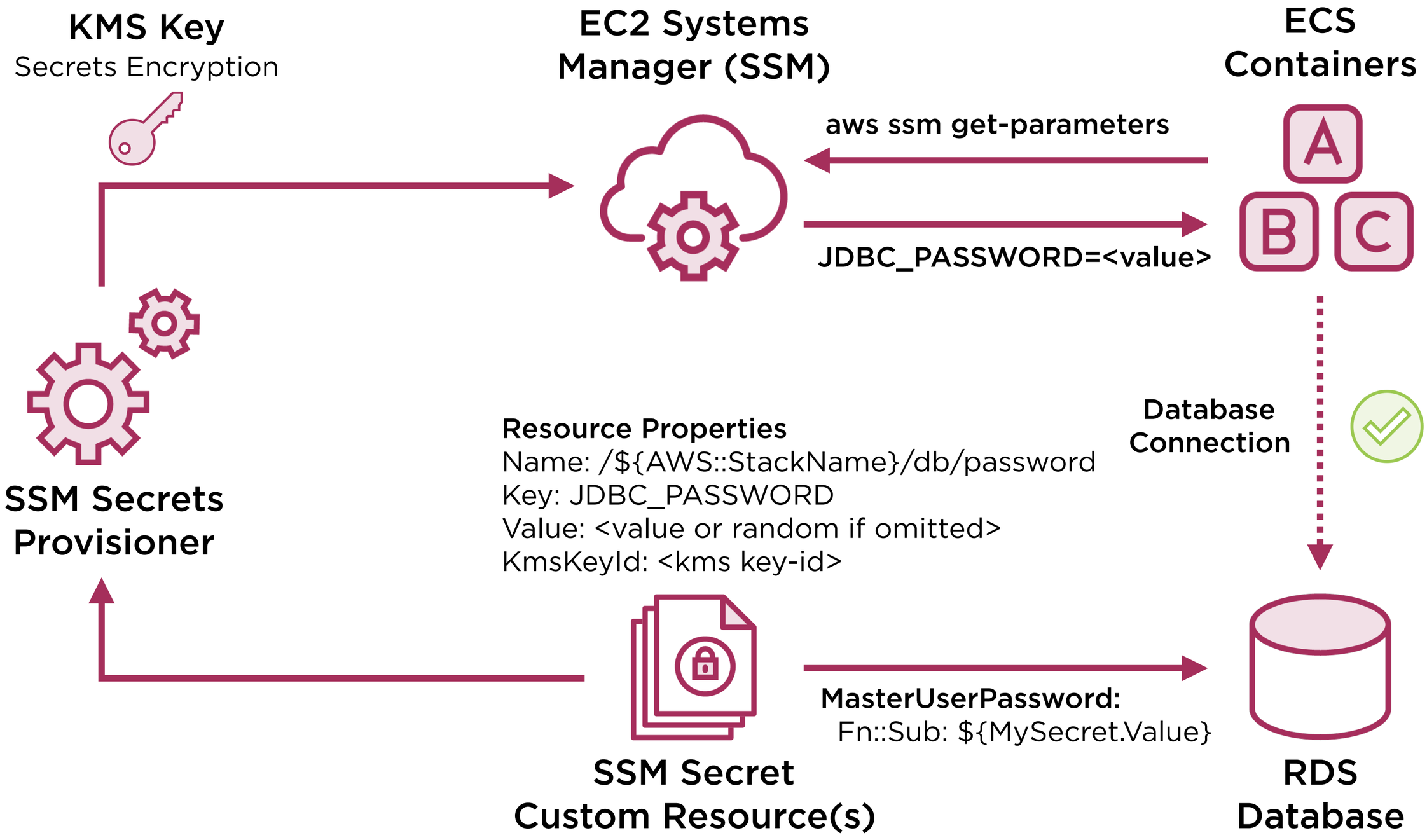
- **Environment Variable Name**
  e.g. JDBC_PASSWORD

`Value: ENCRYPTED_STRING`

- **OPTIONAL**
  Must be encrypted if specified
  Randomly generated if omitted

`KmsKeyId: GUID`

- **KMS Key Identifier**
  Used to encrypt/decrypt value


KMS Key
Secrets Encryption
EC2 Systems Manager (SSM)
ECS Containers
aws ssm get-parameters
JDBC_PASSWORD=<value>
A
B
C
SSM Secrets Provisioner
Resource Properties
Name: /${AWS::StackName}/db/password
Key: JDBC_PASSWORD
Value: <value or random if omitted>
KmsKeyId: <kms key-id>
Database Connection
MasterUserPassword:
Fn::Sub: ${MySecret.Value}
SSM Secret Custom Resource(s)
RDS Database

Creating the Secrets Provisioner

Creating Secrets using CloudFormation

Injecting Secrets at Container Startup

Consuming Secrets In CloudFormation

Resources Consuming Secrets in our Stack

ECS Task Definitions

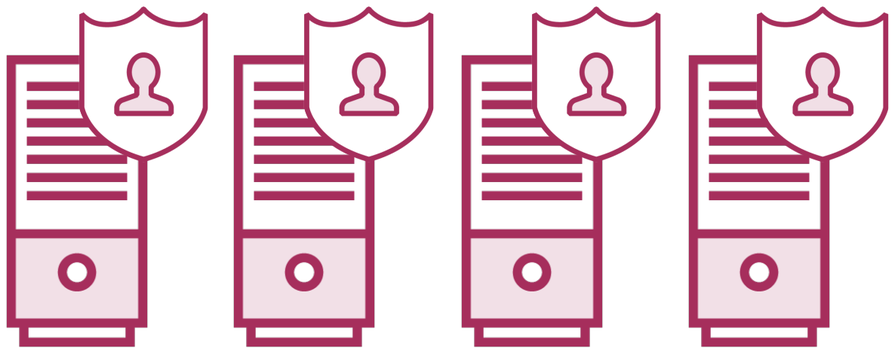
RDS Instance
Master User Password

Configuring IAM Roles for Accessing Secrets

Default IAM Security for ECS Tasks

ECS Task
App A

ECS Task
App B



ECS Container Instances

By default, ECS Tasks inherit privileges of the ECS Container Instance Role

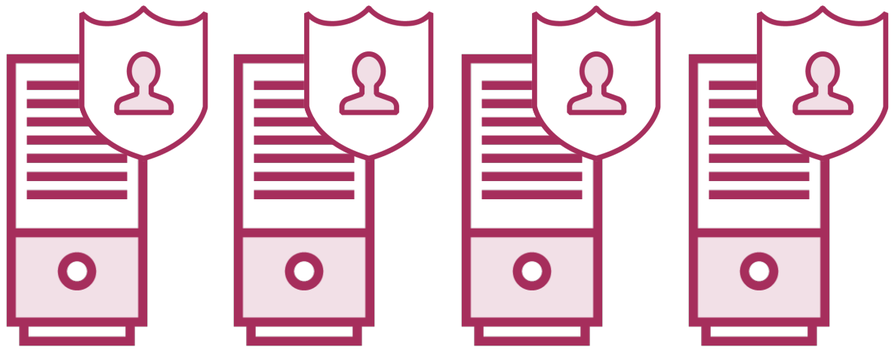
ECS Container Instance Role
EC2 Instance Profile

ECS Task IAM Roles

ECS Tasks will only have privileges assigned to their specific roles

ECS Task
App A

ECS Task
App B



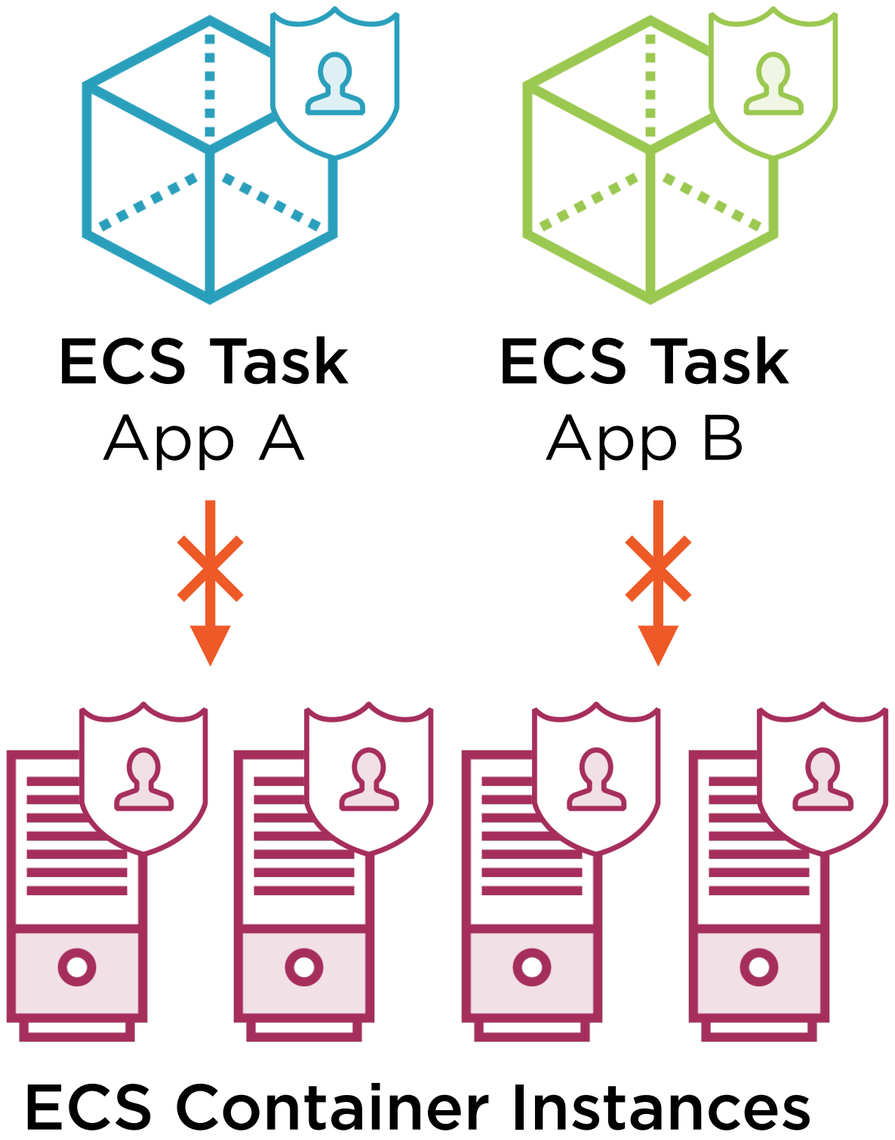
ECS Container Instances

Application A Role
ECS Task Role

Application B Role
ECS Task Role

ECS Container Instance Role
EC2 Instance Profile

ECS Task Roles do NOT inherit the underlying ECS container instance role



Application A Role
ECS Task Role

Application B Role
ECS Task Role

ECS Container Instance Role
EC2 Instance Profile

Deploying Secrets using CloudFormation

Summary

Managing Secrets in AWS

- EC2 Systems Manager
- Key Management Service
- Secrets Provisioner Custom Resource
- Injecting secrets at container startup
- Exposing secrets to other CloudFormation resources