

# Managing Compute Workflows

Beyond basic use-cases & small companies

- ▶ Managing Compute Tasks At Scale
- ▶ Managing Instance & Server Fleets
- ▶ Managing Configuration & Parameters At Scale

# Planning & Performing Batch Jobs



## AWS Batch

### Create Job Definitions

Define Fargate or EC2  
instance jobs

Define image & basic  
hardware requirements

Extra configuration:  
Permissions, file systems, ...

### Execute Jobs

Submit or schedule jobs

AWS provisions resources &  
executes job

Jobs & job status can be  
tracked

# Optimizing Compute Resources



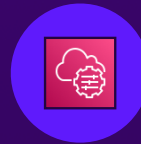
**AWS Compute Optimizer**



Uses machine learning to  
analyze CloudWatch metrics  
& resource configurations

Recommends improvements  
(e.g., to use a different  
instance type or memory  
settings)

# Managing Large Scale Systems



## Systems Manager

A service with many capabilities that help with managing large fleets of servers & applications

### Node Management

Group, visualize & manage a fleet of servers

Connect to servers via Session Manager

Orchestrate patches & server-wide commands

### Operations Management

Manage server-wide operations

Manage incidents

Fleet monitoring

### Application Management

Manage application parameters

Provide & manage application configuration

Easily deploy or roll back configuration changes

### Change Management

Manage fleet changes & updates

Automate change requests

Configure standardized maintenance windows

# Provide Standardized Service Solutions

## Problem

Not every account user should create a custom solution



## Service Catalog

Create standardized, configurable AWS service usage templates

e.g., a VPC with an EC2 instance and a RDS instance

← can be combined



## Proton

Create standardized serverless & container deployments

e.g., a VPC with an ECS cluster on Fargate



## Launch Wizard

Helps with launching standardized, pre-built (by AWS) applications

e.g., launch a SAP application

# Summary



## Size Matters

Micro-management does not work for large-scale cloud usage

Operating & monitoring individual services is not possible

System-wide solutions are needed: Systems Manager etc.



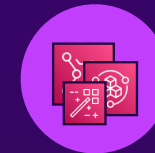
## Optimizing & Managing Compute Resources

**Systems Manager:** Manage server fleets & all applications

Manage updates, incidents or changes globally

Perform batch operations with less effort via **AWS Batch**

Optimize compute usage via **Compute Optimizer**



## Standardizing Applications & Resources

Account users shouldn't create different, custom solutions

Standardized recipes via **Proton** or **Service Catalog**

Pre-built (AWS-managed) apps via **Launch Wizard**