

# LAB: Systems Manager – Parameter Store

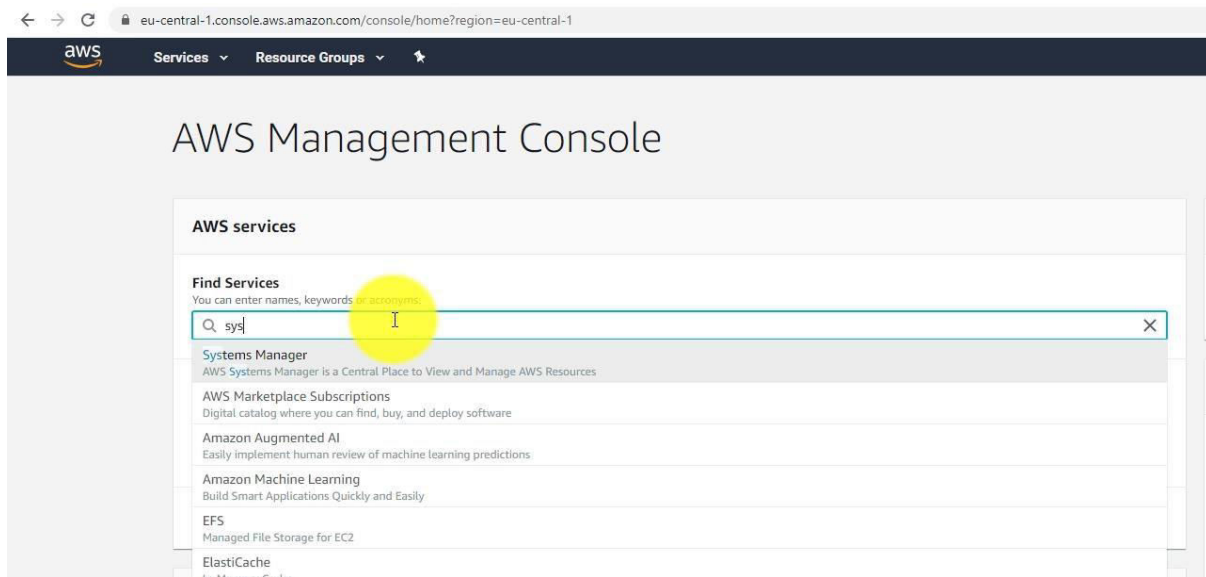
You need:

- An AWS Account

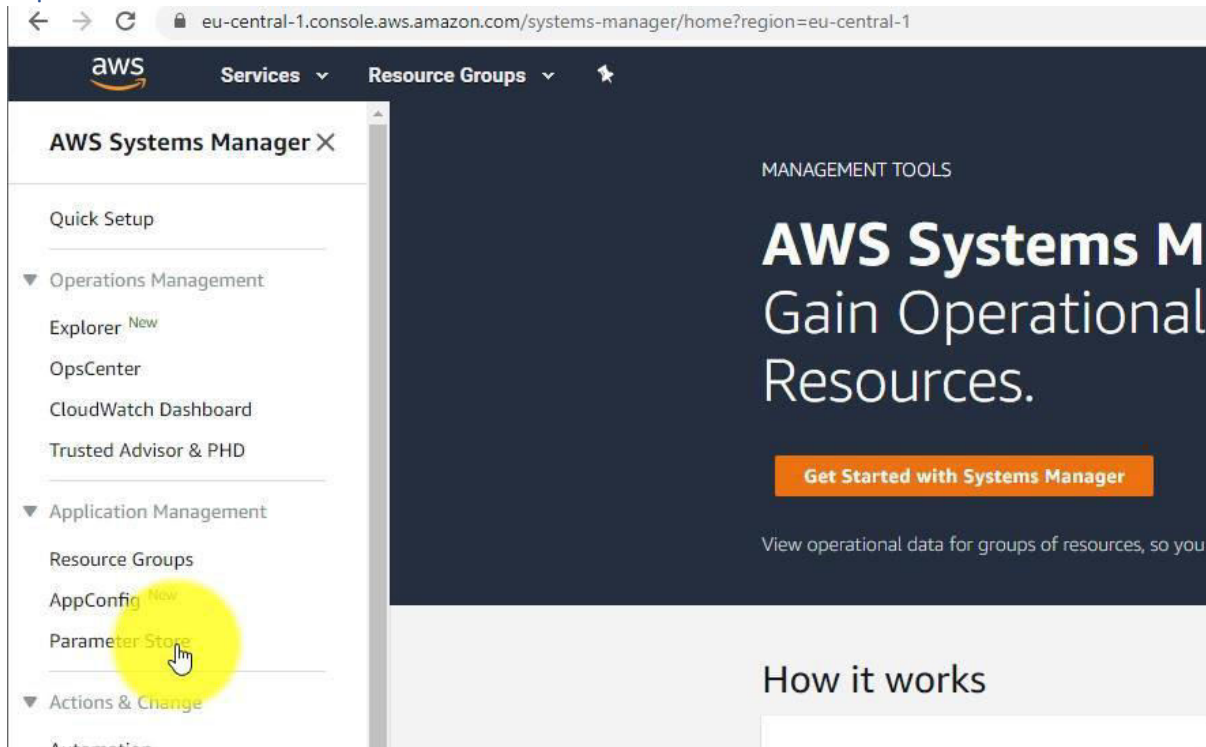
**Duration of the Lab:** 30 Minutes.

**Difficulty:** medium

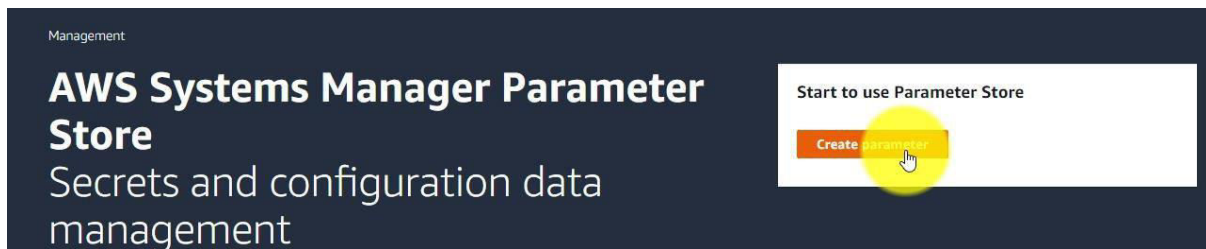
## Open the Systems Manager



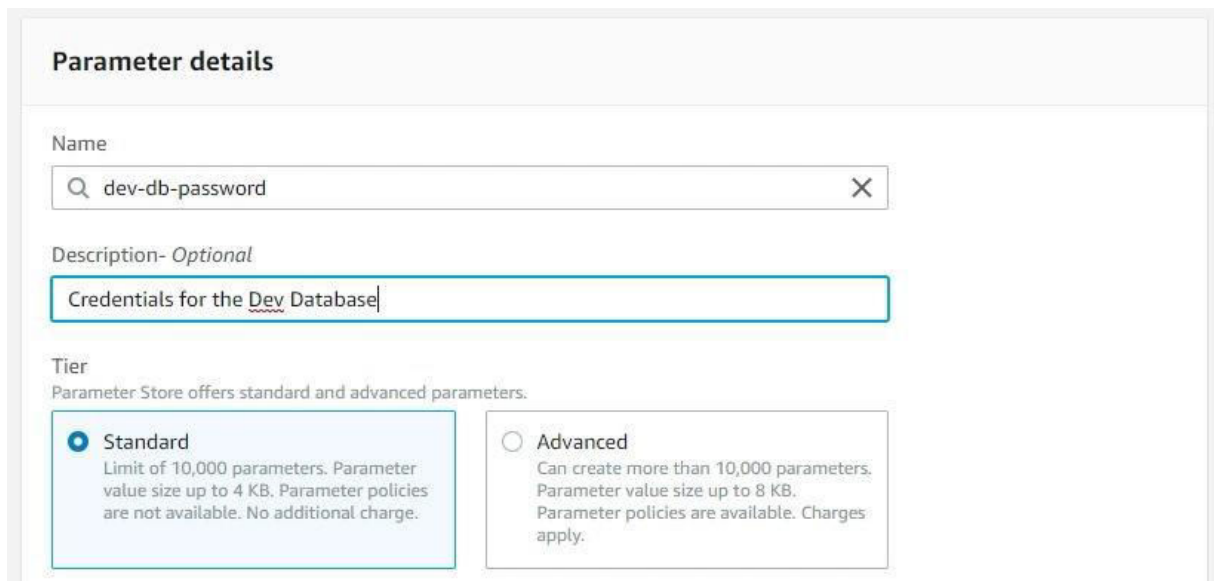
## Open the Parameter Store Feature



## Create a Parameter



Give it a name – for example a tiered name where “dev-\*” are all development parameters and “prod-” are all production parameters. You can later on set corresponding rights in the IAM Permissions we attach to the ECS cluster to read only “dev-” parameters in our dev-cluster:

The image shows the "Parameter details" form in the AWS console. It has a "Name" field with a search icon and the text "dev-db-password", and a "Description- Optional" field with the text "Credentials for the Dev Database". Below these is a "Tier" section with two options: "Standard" (selected with a radio button) and "Advanced". The "Standard" option has a description: "Limit of 10,000 parameters. Parameter value size up to 4 KB. Parameter policies are not available. No additional charge." The "Advanced" option has a description: "Can create more than 10,000 parameters. Parameter value size up to 8 KB. Parameter policies are available. Charges apply."

Select a secure string and enter the master password for *your* RDS Database:

## Complete AWS ECS DevOps Masterclass for Beginners

Type

☐ String  
Any string value.

☐ StringList  
Separate strings using commas.

☒ SecureString  
Encrypt sensitive data using the KMS keys for your account.

KMS key source

☒ My current account  
Use the default KMS key for this account or specify a customer-managed CMK for this account. [Learn more](#)

☐ Another account  
Use a KMS key from a different account. [Learn more](#)

KMS Key ID

alias/aws/ssm ▼

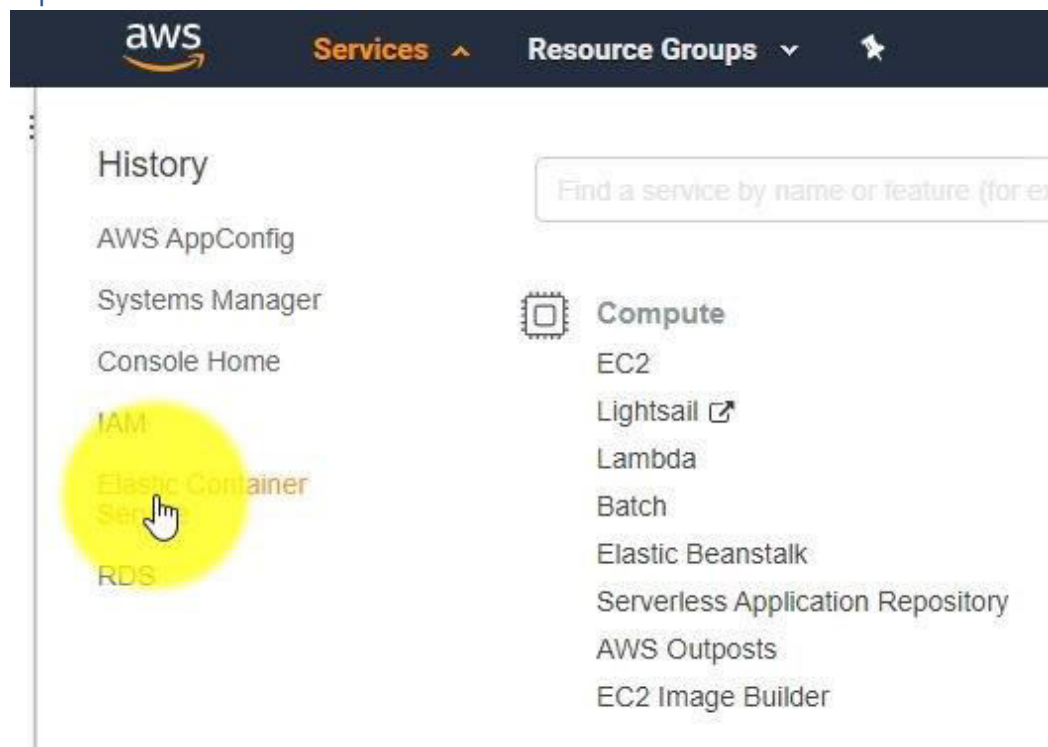
Value

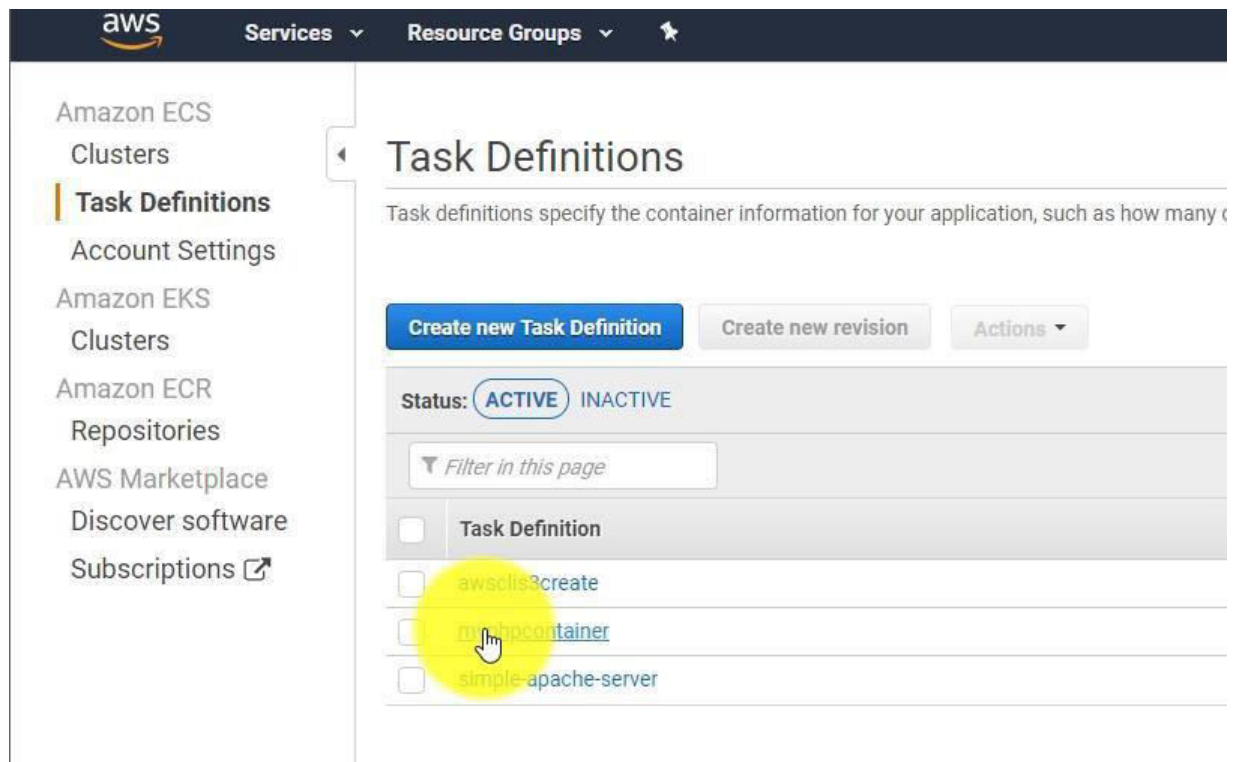
49lFAa92KIHHIT1Xuujd I

Maximum length 4096 characters.

Then simply hit “create parameter”

## Update the ECS Task Definition

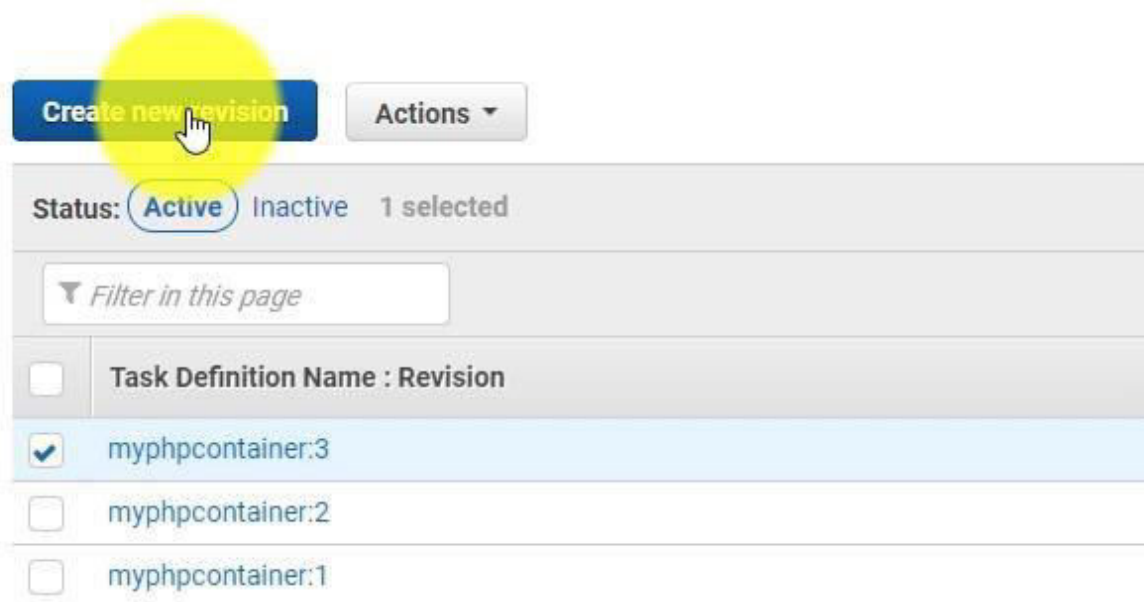




The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', and 'Resource Groups'. The left sidebar lists various services: Amazon ECS (Clusters, Task Definitions, Account Settings), Amazon EKS (Clusters), Amazon ECR (Repositories), AWS Marketplace (Discover software, Subscriptions), and a link to 'Subscriptions'. The main content area is titled 'Task Definitions' and includes a description: 'Task definitions specify the container information for your application, such as how many c'. Below this, there are buttons for 'Create new Task Definition', 'Create new revision', and an 'Actions' dropdown. A status filter shows 'ACTIVE' selected and 'INACTIVE' as an option. A search bar is labeled 'Filter in this page'. A table lists task definitions with checkboxes and names: 'Task Definition', 'awscli3create', 'myphpcontainer' (highlighted with a yellow circle and a mouse cursor), and 'simple-apache-server'.

## Task Definition Name : myphpcontainer

Select a revision for more details



The screenshot shows the 'myphpcontainer' task definition details page. At the top, there is a 'Create new revision' button (highlighted with a yellow circle and a mouse cursor) and an 'Actions' dropdown. Below this, the status is shown as 'Active' (selected) and 'Inactive', with a note '1 selected'. A search bar is labeled 'Filter in this page'. A table lists revisions with checkboxes and names: 'Task Definition Name : Revision', 'myphpcontainer:3' (checked), 'myphpcontainer:2', and 'myphpcontainer:1'.

## Complete AWS ECS DevOps Masterclass for Beginners

The screenshot shows the AWS ECS console interface. On the left, there's a sidebar with options like 'Task memory maximum allocation for container memory reservation', 'Task CPU maximum allocation for containers', 'Container Definitions', and 'Service Integration'. The main area is titled 'Environment variables' and includes a 'Working directory' field set to '/usr/app'. Below this, there's a table for environment variables with columns 'Key' and 'Value'. The 'Key' column has entries for 'DB\_HOST', 'DB\_PASSWORD', and 'DB\_USER'. The 'Value' column has corresponding values: 'database-1.cf4mzi8yclyb.eu-central-1.rds.amazonaws.', '49lFAa92KlHhT1Xuujc', and 'admin'. A yellow circle highlights the 'Value' dropdown menu for 'DB\_PASSWORD'. Below the table, there's a section for 'STARTUP DEPENDENCY ORDERING' with a table for 'Container name' and 'Condition'.

In the environment variables for your container update the password field so it fetches the password from the Parameter Store instead of having it plain-text in the environment variables.

Update to “ValueFrom” from the Dropdown and enter an ARN (Amazon Resource Name) for the dev-db-password from your parameter store. This has the following format:  
arn:aws:ssn:REGION:USERID:parameter/dev-db-password

For my own region (eu-central-1) and my own User-ID it looks like this:

This screenshot shows the 'Environment variables' section of the AWS ECS console. The 'DB\_PASSWORD' entry now has its 'Value' dropdown changed to 'ValueFrom' (indicated by a red circle with the number 1). The 'ValueFrom' dropdown is open, showing a list of parameter store ARNs. The selected ARN is 'arn:aws:ssn:eu-central-1:161952721022:parameter/dev-db-password' (indicated by a red circle with the number 2). The other entries, 'DB\_HOST' and 'DB\_USER', remain unchanged.

Then hit “Update” and create the new TaskDefinition revision.

## Update the IAM Role

The Task itself has an IAM Role attached for the task execution. This role must get access to the Parameter, otherwise ECS can't access it and can't read the parameter and inject it into the environment.

## Complete AWS ECS DevOps Masterclass for Beginners

The screenshot shows the AWS ECS Task Definition console. At the top, there are tabs for 'Builder', 'JSON', and 'Tags', with 'Builder' selected. Below the tabs, the configuration for a task definition named 'myphpcontainer' is displayed. The 'Task Role' is set to 'None'. The 'Network Mode' is set to 'awsvpc'. The 'Compatibilities' are set to 'EC2, FARGATE'. The 'Requires compatibilities' are set to 'FARGATE'. Below these settings, there is a section for 'Task execution IAM role'. It states: 'This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.' A yellow circle highlights the link 'ecsTaskExecutionRole'. Below this, there is a section for 'Task size'.

Create new revision Actions

Builder JSON Tags

Task Definition Name myphpcontainer

Task Role None  
Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the IAM Console.

Network Mode awsvpc  
If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

Compatibilities EC2, FARGATE

Requires compatibilities FARGATE

Task execution IAM role  
This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.

Task execution role [ecsTaskExecutionRole](#)

Task size  
The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container level memory settings for Windows containers.

Click on the role and then attach some inline policy to get the credentials:

As a service choose the Systems Manager:

The screenshot shows the AWS IAM console. At the top, there is a dropdown menu labeled 'Select a service'. Below the dropdown, there is a search bar with the text 'sys' entered. A yellow circle highlights the search bar. Below the search bar, there is a list of services. The first service is 'Systems Manager'.

Select a service

Service Select a service below

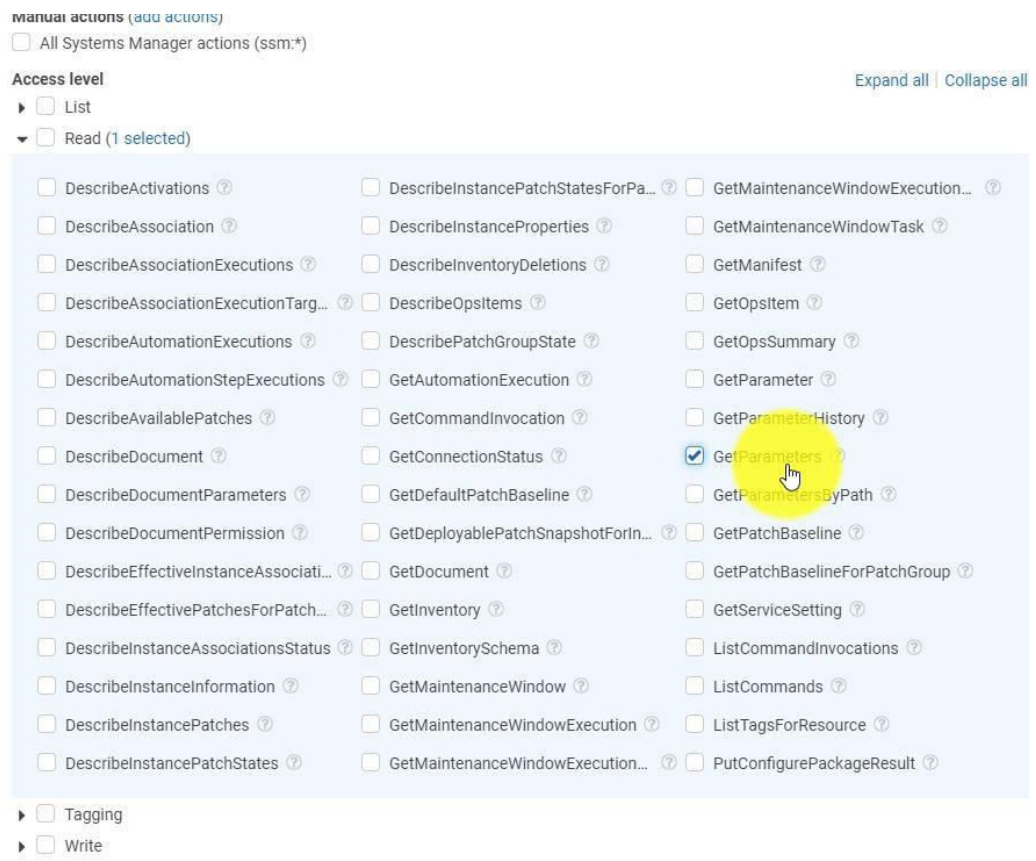
close

Q sys

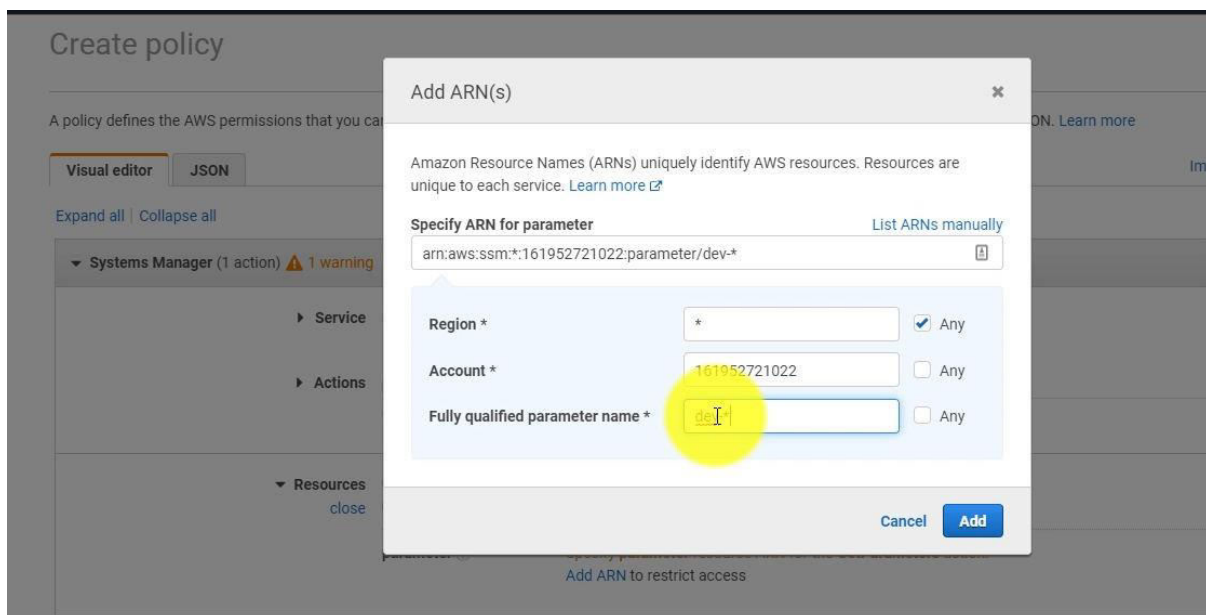
Systems Manager

Then select action: Read, Get Parameters:

## Complete AWS ECS DevOps Masterclass for Beginners



As a resource select all “dev-\*” parameters:



Then give the policy a name and create the policy:



## Complete AWS ECS DevOps Masterclass for Beginners

## Review policy

Before you create this policy, provide the required information and review this policy.

**Name\***

Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.

Maximum 128 characters. Use alphanumeric and '+=, @\_-' characters.

## Summary

<input type="text" value="Filter"/>			
Service	Access level	Resource	Request condition
Allow (1 of 226 services) <a href="#">Show remaining 225</a>			
Systems Manager	Limited: Read	arn:aws:ssm:*:161952721022:paramet er/dev-*	None



\* Required

Cancel

[Previous](#)

Create policy

## Update the Service

Now, in order to take effect, you have to update the service to your new TaskDefinition revision:

Go to your Fargate Cluster and select the php service:

Amazon ECS

Clusters

Task Definitions

Account Settings

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions ↗

Services ▾

Resource Groups ▾

🏠

Clusters > fargatecluster

Cluster : fargatecluster

Get a detailed view of the resources on your cluster.

Cluster ARNarn:aws:ecs:eu-central-1:161952721022:cluster/fargatecluster

StatusACTIVE

Registered container instances0

Pending tasks count0 Fargate, 0 EC2

Running tasks count1 Fargate, 0 EC2

Active service count1 Fargate, 0 EC2

Draining service count0 Fargate, 0 EC2

ServicesTasksECS InstancesMetricsScheduled TasksTagsCapacity Providers

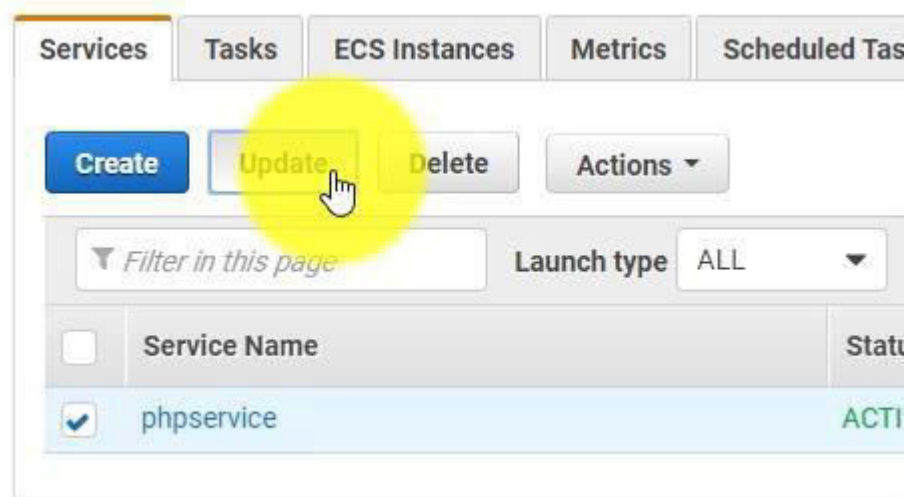
CreateUpdateDeleteActions ▾

Filter in this pageLaunch typeALLService typeALL1 selected

<input type="checkbox"/>	Service Name	Status	Service type	Task Definition
<input checked="" type="checkbox"/>	myphpservice	ACTIVE	REPLICA	myphpcontainer:3



And then hit update:



And select the latest revision from the dropdown:

### Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.



Force new deployment and run through the wizards:

## Complete AWS ECS DevOps Masterclass for Beginners

Force new deployment ☒ ⓘ

Cluster  ⓘ

Service name  ⓘ

Service type\* REPLICA ⓘ

Number of tasks  ⓘ

Minimum healthy percent  ⓘ

Maximum percent  ⓘ

[Cancel](#) [Skip to review](#) [Next step](#)

Wait until the new task comes up:

Details	Tasks	Events	Auto Scaling	Deployments	Metrics	Tags	Logs
Last updated on April 10, 2020 11:00:24 AM							
Task status: <span>Running</span> Stopped							
<input type="text" value="Filter in this page"/> < 1-2 >							
Task	Task Definition	Last status	Desired status	Group	Launch type	Platform v	
17a35758-c47e-40a0-8761-b851...	myphpcontainer:5	PROVISIONING	RUNNING	service:phpservice	FARGATE	1.3.0	
92236a1c-077f-44c4-9271-7b12...	myphpcontainer:3	RUNNING	RUNNING	service:phpservice	FARGATE	1.3.0	

Now it's up:

Details	Tasks	Events	Auto Scaling	Deployments	Metrics	Tags	Logs
Last updated on April 10, 2020 11:01:04 AM							
Task status: <span>Running</span> Stopped							
<input type="text" value="Filter in this page"/> < 1-2 >							
Task	Task Definition	Last status	Desired status	Group	Launch type	Platform v	
17a35758-c47e-40a0-8761-b851...	myphpcontainer:5	RUNNING	RUNNING	service:phpservice	FARGATE	1.3.0	
92236a1c-077f-44c4-9271-7b12...	myphpcontainer:3	RUNNING	RUNNING	service:phpservice	FARGATE	1.3.0	

Check if the variables was injected properly:

Copy the IP address and open it in a new tab.

Clusters > fargatecluster > Task: 17a35758-c47e-40a0-8761-b85199df0075

## Task : 17a35758-c47e-40a0-8761-b85199df0075

**Details** Tags Logs

**Cluster** fargatecluster

**Launch type** FARGATE

**Platform version** 1.3.0

**Task definition** myphpcontainer:5

**Group** service:phpservice

**Task role** None

**Last status** RUNNING

**Desired status** RUNNING

**Created at** 2020-04-10 11:00:19 +0200

**Started at** 2020-04-10 11:00:59 +0200

### Network

**Network mode** awsvpc

**ENI Id** eni-08c075ebd9c5d3675

**Subnet Id** subnet-d5f6eca8

**Private IP** 172.31.35.23

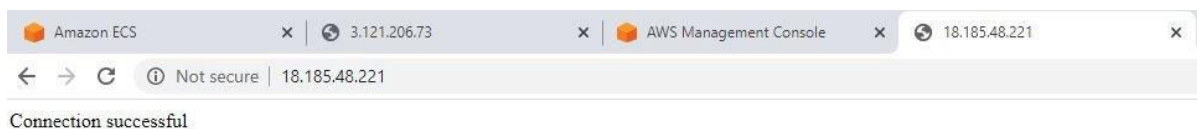
**Public IP** 18.185.48.221

**Mac address** 06:f3:73:63:5c:20

### Containers

Name	Container Runtime ID ...	Status	Image
myphpcont...	218ae618dee16a3bb8...	RUNNING	161952721022.dkr.ecr.eu-central-1....

It should still show Connection successful:



---

*Lab End*

---