LAB: Systems Manager – Parameter Store

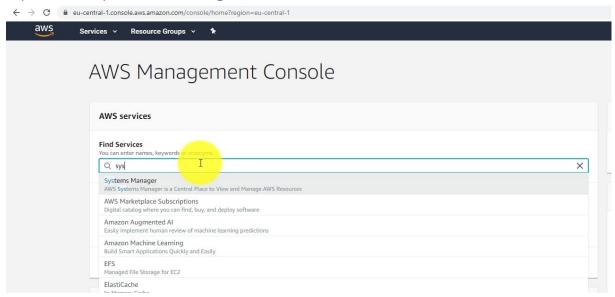
You need:

An AWS Account

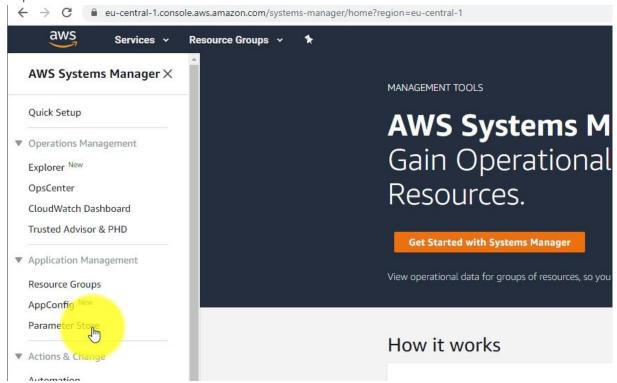
Duration of the Lab: 30 Minutes.

Difficulty: medium

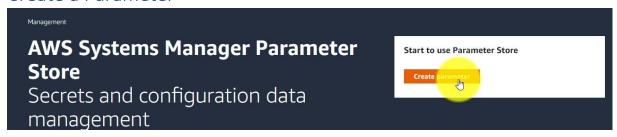
Open the Systems Manager



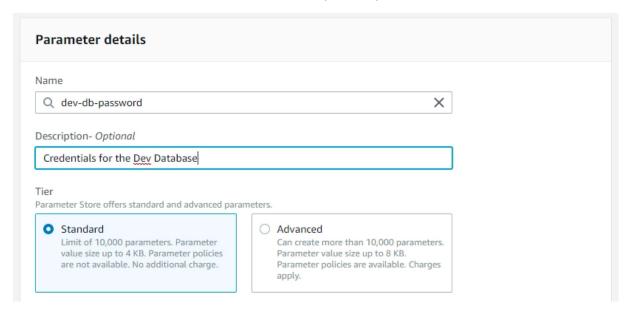
Open the Parameter Store Feature



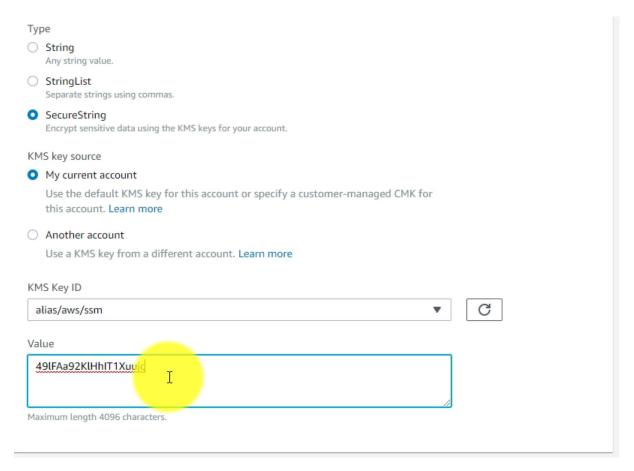
Create a Parameter



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:

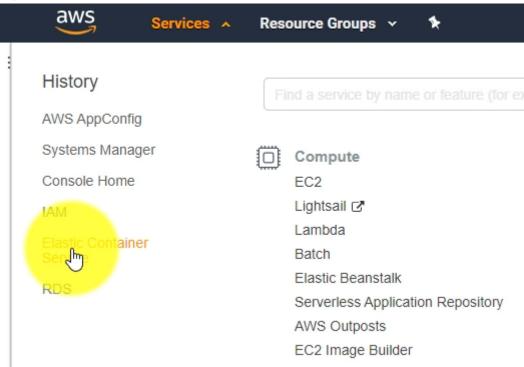


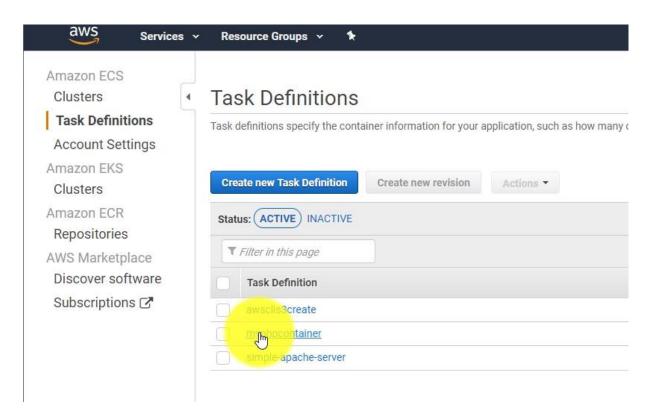
Select a secure string and enter the master password for your RDS Database:



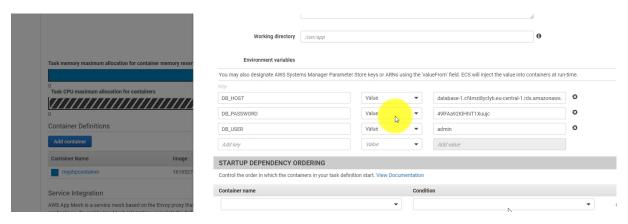
Then simply hit "create parameter"

Update the ECS Task Definition





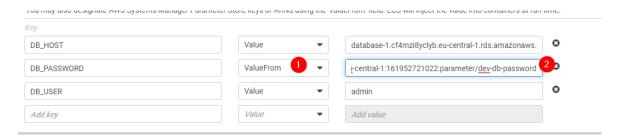
Task Definition Name: myphpcontainer



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 devdb-password from your parameter store. This has the following format: arn:aws:ssn:REGION:USERID:parameter/dev-db-password

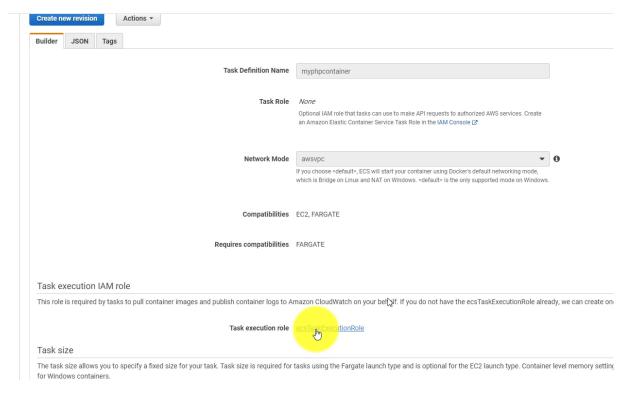
For my own region (eu-central-1) and my own User-ID it looks like this:



Then hit "Update" and create the new TaskDefinition revision.

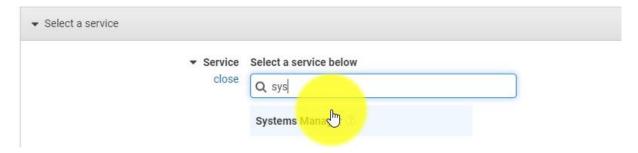
Update the IAM Role

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.

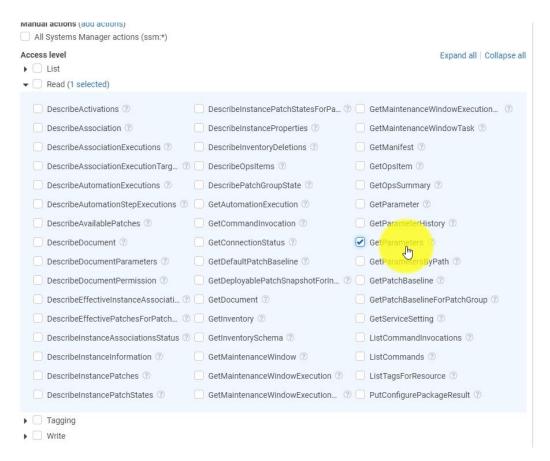


Click on the role and then attach some inline policy to get the credentials:

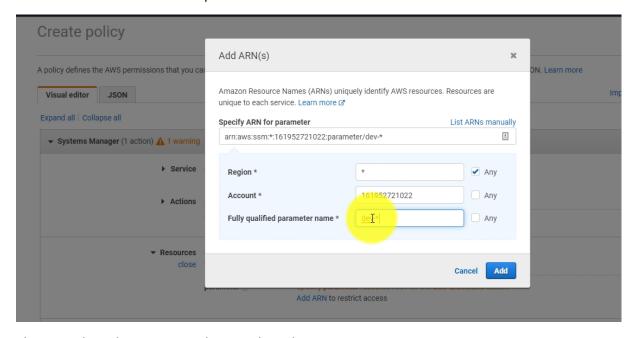
As a service choose the Systems Manager:



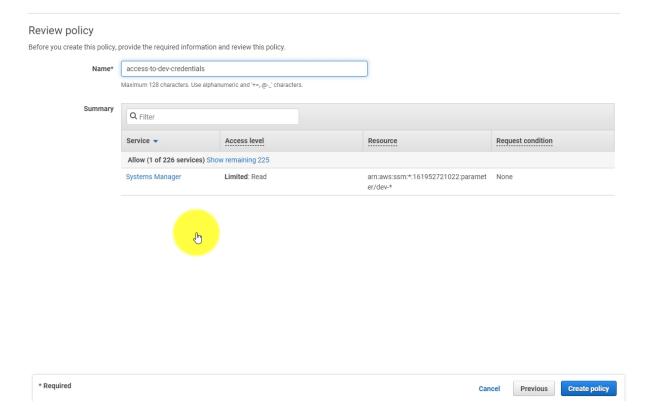
Then select action: Read, Get Parameters:



As a resource select all "dev-*" parameters:



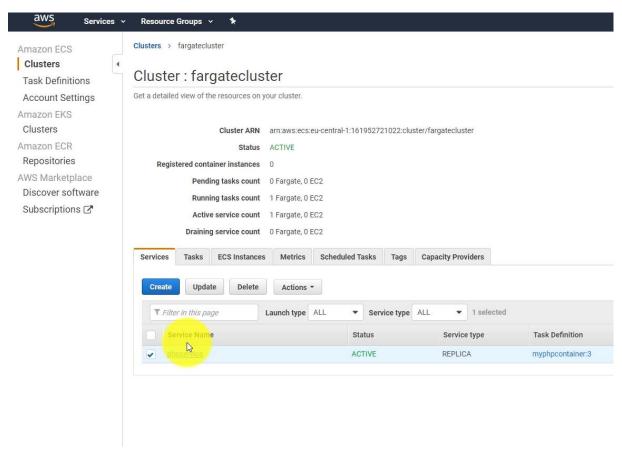
Then give the policy a name and create the policy:



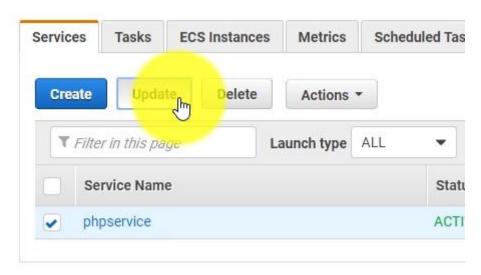
Update the Service

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:



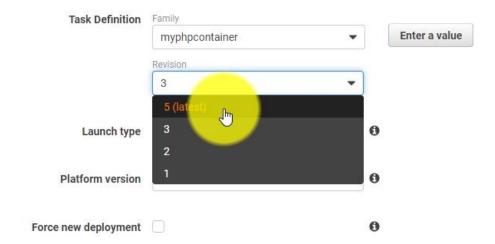
And then hit update:



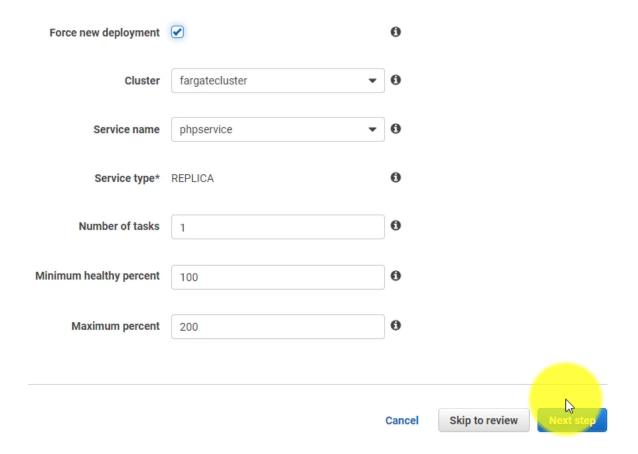
And select the latest revision from the dropdown:

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 E Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number tasks in your service.



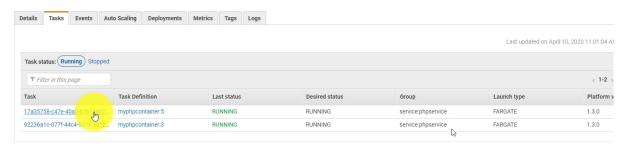
Force new deployment and run through the wizards:



Wait until the new task comes up:



Now it's up:

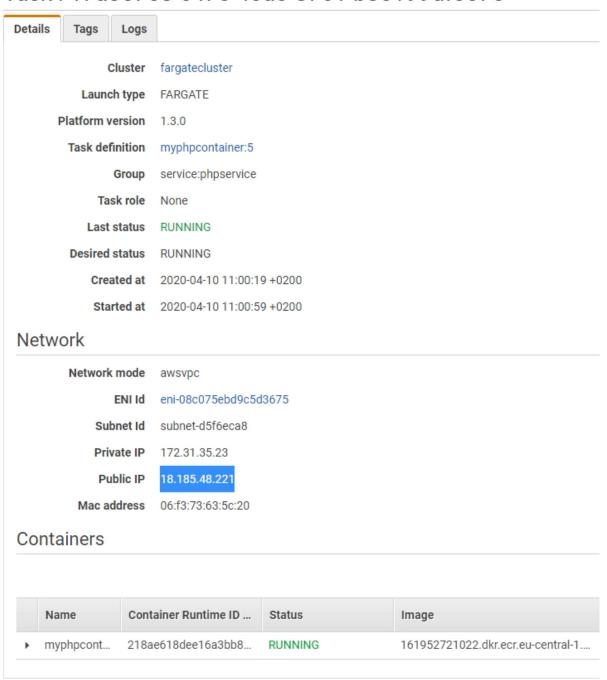


Check if the variables was injected properly:

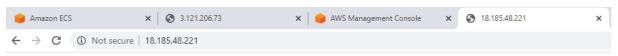
Copy the IP address and open it in a new tab.

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

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



It should still show Connection successful:



Connection successful

Complete	AWS ECS DevOps Masterclass for Beginners
	Lah End