**Domain: Container Services**

**Topic: ECS**

**Task: Create nginx service on ECS Fargate**

**Problem to Be Solved**

*Containeraized service creation without time wastes on underline infrastructure resources.*

## Explanation of the Solution

*Service consisting from 1 task run on Fargate type of ECS*

## Implementation Details

*Prerequisites:*

1. *Repository and image from ECR practical task*
2. *Appropriate IAM polices -* ***AmazonECS\_FullAccess*** *AWS managed policy**might be assigned for this task.*
3. *VPC subnet where you will place ECS Service and subnets for your loadbalancer (at least 2).*

*LoadBanalcer creation:*

1. *In* ***EC2->Load Balancer*** *section of AWS console use* ***Create Load Balancer*** *button and choose Application Load Balancer.*
2. *On* ***Step 1: Configure Load Balancer*** *enter* ***Name*** *for your LoadBalancer, ensure that we have* ***internet-facing*** *scheme selected.*
3. *In* ***Availablility Zones*** *section choose at least 2 subnets enabling appropriate checkboxes (these subnets are for loadbalancers not for ECS placement!) and click Next moving to other configuration tabs*
4. *On* ***Step3: Configure Security******Groups*** *choose creation of a new security group (I suggest adjust name to* ***ecs-task-lb-sg****) and select* ***HTTP*** *type of request to* ***80*** *port from* ***My IP*** *Source.*
5. *On* ***Step 4: Configure Routing*** *tab keep default* ***New target group*** *in Target group**field; give a* ***Name*** *and select* ***IP*** *in* ***Target type***  *setting.*
6. *Skip* ***Register Targets*** *tab – we well attach target on ECS side and finish LB configuration on* ***Review*** *tab clicking* ***Create*** *button*

*Task Definition:*

1. *Go* ***ECS->Task Definition*** *and press* ***Create new Task Definition*** *button.*
2. *Select FARGATE launch type on the appeared window as step1.*
3. *On Configure task and container definitions page enter* ***Name****; in* ***Task Size*** *section choose 0.5GB and 0.25 vCPU for Task memory (GB) and Task CPU (vCPU) .*
4. *In* ***Container Definitions*** *press* ***Add container*** *button.*
5. *In pop-up windows enter* ***Container Name*** *– nginx;* ***Image*** *should refer to the image uploaded in ECR task part – it should have a url in following format – {account\_id}.dkr.ecr.{region}.amazonaws.com/ecr-task-{last-name}:nginx;  
   add* ***80*** *port to* ***Port mappings*** *setting.  
   The rest of settings should be left as is – just press* ***Add*** *button to return to Task Definition configuration.*
6. *Complete Task Definition creation pressing* ***Create*** *button.*

*ECS Cluster:*

1. *Go* ***ECS->Cluster*** *and press* ***Create Cluster*** *button.*
2. *Select* ***Networking only*** *cluster template type on the appeared window as step1.*
3. *On Configure cluster page enter* ***Name*** *and press* ***Create*** *button.*

*ECS Service:*

1. *On* ***ECS->Cluster*** *page select you cluster created at d3 step and click on it.*
2. *Press* ***Create*** *button on* ***Services*** *tab*
3. *Choose* ***Launch type*** *–* ***FARGATE****; ensure that Tast Definition is created in c5 step previously; enter* ***Service Name****; specify* ***1*** *in* ***Nubmer of tasks*** *field and press* ***Next Step*** *button.*
4. *In* ***Cluster VPC*** *select VPC where you want place your ECS computing resources; choose appropriate* ***Subnets*** *and* ***Security groups*** *(I allow access only from loadbalancer’s CIDR ranges to HTTP 80 port).*
5. *Select* ***Application Load Balancer*** *as type of loadbalancing and find loadbalancer created by b6 step earlier.*
6. *Click on* ***Add to loadbalancer*** *button to add your container – there should additional configuration fields appear.*
7. *In* ***Production listener port\**** *select already existing* ***80:HTTP;*** *in* ***Target group name*** *field select target group created by b5 Step earlier.*
8. *The rest of settings keep without modifications and click* ***Next*** *button few times and* ***Create Servic****e to complete.*

*Verification:*

1. *Find out DNS name of your Loadbalancer and paste it to the browser – you should see default Nginx welcome page.*

*(Extra)Container modification:*

1. *Add to your Dockerfile from ECR practical task line to change background of default Nginx welcome page*

RUN sed -i '/body {/a background-color: #E6E6FA;' /usr/share/nginx/html/index.html

1. *Build new image* docker build --file Dockerfile --rm --tag {account\_id}.dkr.ecr.{region}.amazonaws.com/ecr-task-{last\_name}:nginx-changed .
2. *Push the image to the ECR:*aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin {account\_id}.dkr.ecr.{region}.amazonaws.com

docker push {account\_id}.dkr.ecr.{region}.com/ecr-task-{last\_name}:nginx-changed

*(Extra) Update service with new image:*

1. *On* ***Task Definition*** *tab select task created by c6 Step earlier and press* ***Create new revision*** *button.*
2. *Scroll down till* ***Container Definitions*** *section and remove current* ***nginx*** *container clicking on cross sign.*
3. *Press* ***Add Container*** *button. Fill the name and select newly uploaded nginx -* {account\_id}.dkr.ecr.{region}.com/ecr-task-{last\_name}:nginx-changed  
   *Don’t forget add 80 port to mapping.*
4. *Update your service to the latest version of Task Definition via* ***Update Service*** *button*
5. *Stop previous task on* ***Tasks*** *tab selecting task definition tailing with :1 version via* ***Stop*** *button and observe the result in browser.*

## Benefits / Outcomes / Pros and Cons / Summary

*We don’t waste time on maintenance underlying hardware – it’s headache of AWS.*

## Tearing down

*Destroy Loadbalancer (including target group), Tasks Definitions, Cluster, ECR stuff and SG’s*