

Using Docker on AWS with the CLI

aws, ecs-cli – install from : https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_CLI_installation.html

configuring: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_CLI_Configuration.html

ecs-cli

configuring all on the command line

```
ecs-cli configure --cluster ec2-test-App --region us-east-1 --default-launch-type EC2 --config-name ec2-test-App
ecs-cli configure --cluster ec2-test-App --region eu-central-1a --default-launch-type EC2 --config-name ec2-test-App
```

configure a Cluster

```
ubuntu@ubuntu:~$ ecs-cli configure \
> --cluster ec2-test-App \
> --region us-east-1 \
> --default-launch-type EC2 \
> --config-name ec2-test-App
INFO[0000] Saved ECS CLI cluster configuration ec2-test-App.
```

```
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ ecs-cli configure profile \
> --access-key AKIAIQIOFRRYP7PLTN7Q \
> --secret-key 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 \
> --profile-name ec2-test-App
```

create a default and
ec2-test-App profile

```
ecs-cli configure profile --profile-name profile_name --access-key $AWS_ACCESS_KEY_ID --secret-key $AWS_SECRET_ACCESS_KEY
```

```
ecs-cli configure profile --profile-name ec2-test-App --access-key AKIA5MDFWA...YVQN7 ¥
--secret-key YGGDh81vP1BzNrZ+L5nN2yt.....0n9q8KFAdJcj
```

```
$ ecs-cli configure profile --access-key AKIA5MDFWAMC...YVQN7 --secret-key YGGDh81vP1BzNrZ+L5nN2ytz----0n9q8KFAdJcj
INFO[0000] Saved ECS CLI profile configuration default.
```



```
$ cat /Users/luigicavuoti/.ecs/credentials
version: v1
default: default
ecs_profiles:
  default:
    aws_access_key_id: AKIA5MDFWAM---YVQN7
    aws_secret_access_key: YGGDh81vP1BzNrz+L5nN2ytz---0n9q8KFAJcj
ec2-test-App:
  aws_access_key_id: AKIA5MDFWAMC3EUYYVN7
  aws_secret_access_key: YGGDh81vP1BzNrz+L5nN2ytz---0n9q8KFAJcj
```



ecs-cli profile saved

```
$ cat /Users/luigicavuoti/.ecs/config
version: v1
default: ec2-test-App
clusters:
  ec2-test-App:
    cluster: ec2-test-App
    region: us-east-1
    default_launch_type: EC2
```



ecs-cli config stored here!

Running the instance

```
ecs-cli up --capability-iam --size 2 --instance-type t2.medium --cluster-config ec2-test-App --ecs-profile ec2-test-App
```

```
$ ecs-cli up --capability-iam --size 2 --instance-type t2.medium --cluster-config ec2-test-App --ecs-profile ec2-test-App
WARN[0000] You will not be able to SSH into your EC2 instances without a key pair.
INFO[0002] Using recommended Amazon Linux 2 AMI with ECS Agent 1.52.0 and Docker version 19.03.13-ce
INFO[0003] Created cluster                                cluster=ec2-test-App region=us-east-1
INFO[0005] Waiting for your cluster resources to be created...
INFO[0005] Cloudformation stack status                    stackStatus=CREATE_IN_PROGRESS
INFO[0067] Cloudformation stack status                    stackStatus=CREATE_IN_PROGRESS
INFO[0128] Cloudformation stack status                    stackStatus=CREATE_IN_PROGRESS
```

```
VPC created: vpc-0217fae613b3de37b
Security Group created: sg-06d6d831df74e65a8
Subnet created: subnet-04ff22c25b984bbea
Subnet created: subnet-04a79661ad9357b64
Cluster creation succeeded.
```

shows object id for
each of the resources,
that we created!

running the app with up command
which config to run for which
cluster?

```
$ ecs-cli up --help
```

```
NAME:
  ecs-cli up - Creates the ECS cluster (if it does not already exist) and the AWS resources required to set up the cluster.
```

```
USAGE:
  ecs-cli up [command options] [arguments...]
```

OPTIONS:

```
--capability-iam      Acknowledges that this command may create IAM resources. Required if --instance-role is not
specified. NOTE: Not applicable for launch type FARGATE or when creating an empty cluster.
--empty, -e           [Optional] Specifies that an ECS cluster will be created with no resources.
--instance-role value [Optional] Specifies a custom IAM Role for instances in your cluster. A new instance profile will be
created and attached to this role. Required if --capability-iam is not specified. NOTE: Not applicable for launch type FARGATE.
```

encryption key-pair for ssh conn,
hier not

--keypair value [Optional] Specifies the name of an existing Amazon EC2 key pair to enable SSH access to the EC2 instances in your cluster. Recommended for EC2 launch type. NOTE: Not applicable for launch type FARGATE.

--instance-type value [Optional] Specifies the EC2 instance type for your container instances. If you specify the A1 instance family, the ECS optimized arm64 AMI will be used, otherwise the x86 AMI will be used. Defaults to t2.micro. NOTE: Not applicable for launch type FARGATE.

we wont do anything whith out Docker Container that we created!

we need two files here in our local directory! docker-compose.yml and ecs-params.yml !!!

what should run we will add with docker-compose.yml

```
version: '3'
```

```
services:
```

```
  wordpress:
```

```
    image: wordpress
```

```
    ports:
```

```
      - "80:80"
```

```
    links:
```

```
      - mysql
```

```
  mysql:
```

```
    image: mysql:5.7
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD:
```

Version only till 3! (not over)
we define two services running ,
wordpress and mysql

instead defining the secrets hier we
will export it !

```
$ export MYSQL_ROOT_PASSWORD=bbw123
```

we export the secret on the command line and
hopefully the ecs-cli will pick up the pw and pass it in

the other param file ecs-params.yml we define hier:

```
version: 1
```

```
task_definition:
```

```
services:

wordpress:

  cpu_shares: 100

  mem_limit: 524288000

mysql:

  cpu_shares: 100

  mem_limit: 524288000
```

startet a new pair of instances

```
$ ecs-cli up --capability-iam --size 2 --instance-type t2.medium --cluster-config ec2-test-App --ecs-profile ec2-test-App
WARN[0000] You will not be able to SSH into your EC2 instances without a key pair.
INFO[0003] Using recommended Amazon Linux 2 AMI with ECS Agent 1.52.0 and Docker version 19.03.13-ce
INFO[0003] Created cluster                                cluster=ec2-test-App region=us-east-1
INFO[0005] Waiting for your cluster resources to be created...
INFO[0006] Cloudformation stack status                  stackStatus=CREATE_IN_PROGRESS
INFO[0067] Cloudformation stack status                  stackStatus=CREATE_IN_PROGRESS
INFO[0129] Cloudformation stack status                  stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-000d044adb29f312f
Security Group created: sg-03171c734df137ad6
Subnet created: subnet-04f7b56de89229616
Subnet created: subnet-016cfea96f78c49d9
Cluster creation succeeded.
```

shows object id for
each of the resources,
that we created!



EC2-Dashboard New

Ereignisse

Tags

Beschränkungen

▼ Instances

Instances New

Instance-Typen

Instances (4) [Info](#)

<input type="checkbox"/>	Name ▾	Instance-ID	Instance-Status ▾	Instance-Typ ▾	Statusüberprüf...	Alarmstatus	Availability Zone ▾	Öffentlicher I
<input type="checkbox"/>	ECS Instance - ...	i-0da2f8e771df01f05	⊖ Beendet 🔍	t2.medium	–	Keine Ala... +	us-east-1a	–
<input type="checkbox"/>	ECS Instance - ...	i-0a3127c8803d68f69	✅ Läuft 🔍	t2.medium	⌚ Wird initialisiert	Keine Ala... +	us-east-1a	ec2-184-72-1
<input type="checkbox"/>	ECS Instance - ...	i-0042c8732f9fb98bf	⊖ Beendet 🔍	t2.medium	–	Keine Ala... +	us-east-1b	–
<input type="checkbox"/>	ECS Instance - ...	i-0e150cfaa803fee2f	✅ Läuft 🔍	t2.medium	⌚ Wird initialisiert	Keine Ala... +	us-east-1b	ec2-54-163-6

we just started a new pair of instances now!

Compose

we now start the compose file with the params!

```
version: '3'
services:
  wordpress:
    image: wordpress
    ports:
      - "80:80"
    links:
      - mysql
  mysql:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD:
```

we define two services running ,
wordpress and mysql

instead defining the secrets hier we
will export it !

```
$ export MYSQL_ROOT_PASSWORD=bbw123
```

we export the secret on the command line and
hopefully the ecs-cli will pick up the pw and pass it in

the other param file ecs-params.yml we define hier:

```
version: 1
task_definition:
  services:
    wordpress:
      cpu_shares: 100
      mem_limit: 524288000
```

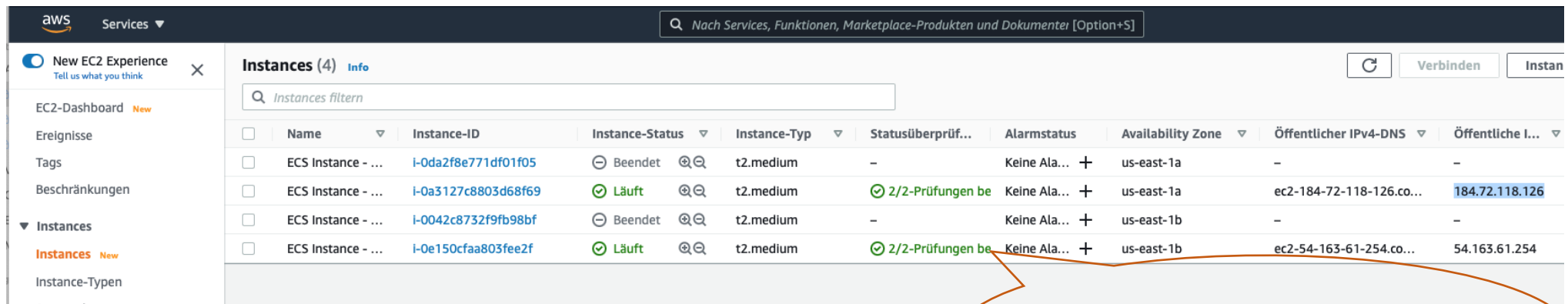


```
mysql:
  cpu_shares: 100
  mem_limit: 524288000
```

start compose-launch the application

```
$ ecs-cli compose --project-name ec2-project service up --cluster-config ec2-test-App
INFO[0001] Using ECS task definition          TaskDefinition="ec2-project:1"
INFO[0001] Auto-enabling ECS Managed Tags
INFO[0012] (service ec2-project) has started 1 tasks: (task 7ae411f2cf334c46957c31ea2d93f5ce). timestamp="2021-05-15 16:43:28 +0000 UTC"
INFO[0043] Service status                    desiredCount=1 runningCount=1 serviceName=ec2-project
INFO[0043] ECS Service has reached a stable state desiredCount=1 runningCount=1 serviceName=ec2-project
INFO[0043] Created an ECS service            service=ec2-project taskDefinition="ec2-project:1"
```

we start the docker containers with ,
we name the project using project-
name!



	Name	Instance-ID	Instance-Status	Instance-Typ	Statusüberprüf...	Alarmstatus	Availability Zone	Öffentlicher IPv4-DNS	Öffentliche I...
<input type="checkbox"/>	ECS Instance - ...	i-0da2f8e771df01f05	Beendet	t2.medium	-	Keine Ala...	us-east-1a	-	-
<input type="checkbox"/>	ECS Instance - ...	i-0a3127c8803d68f69	Läuft	t2.medium	2/2-Prüfungen be	Keine Ala...	us-east-1a	ec2-184-72-118-126.co...	184.72.118.126
<input type="checkbox"/>	ECS Instance - ...	i-0042c8732f9fb98bf	Beendet	t2.medium	-	Keine Ala...	us-east-1b	-	-
<input type="checkbox"/>	ECS Instance - ...	i-0e150cfaa803fee2f	Läuft	t2.medium	2/2-Prüfungen be	Keine Ala...	us-east-1b	ec2-54-163-61-254.co...	54.163.61.254

we start the docker-compose services
with the params inside the 2 instances!

running updates on the container

just edit the compose and params files and rerun

```
$ ecs-cli compose --project-name ec2-project service up --cluster-config ec2-test-App
```

```
version: 1
task_definition:
  services:
    wordpress:
      cpu_shares: 100
      mem_limit: 524288000
    mysql:
      cpu_shares: 100
      mem_limit: 524288000
```

```
version: '3'
services:
  wordpress:
    image: wordpress
    ports:
      - "81:80"
    links:
      - mysql
  mysql:
    image: mysql:5.7
```

environment:

MYSQL_ROOT_PASSWORD:

changing the port!

```
$ ecs-cli compose --project-name ec2-project service up --cluster-config ec2-test-App
$ ecs-cli compose --project-name ec2-project service up --cluster-config ec2-test-App
INFO[0001] Using ECS task definition TaskDefinition="ec2-project:2"
INFO[0001] Updated the ECS service with a new task definition. Old containers will be stopped automatically, and replaced with new ones
desiredCount=1 force-deployment=false service=ec2-project
INFO[0001] Service status desiredCount=1 runningCount=1 serviceName=ec2-project
INFO[0011] (service ec2-project) has started 1 tasks: (task cbf9dec9aec8457a831fe1117a0fe2dc). timestamp="2021-05-15 20:26:43 +0000 UTC"
INFO[0042] Service status desiredCount=1 runningCount=2 serviceName=ec2-project
INFO[0084] Service status desiredCount=1 runningCount=1 serviceName=ec2-project
INFO[0084] (service ec2-project) has stopped 1 running tasks: (task a8a4e3e913a047989928a5a9aa7eebdd). timestamp="2021-05-15 20:27:56 +0000 UTC"
INFO[0094] ECS Service has reached a stable state desiredCount=1 runningCount=1 serviceName=ec2-project
```

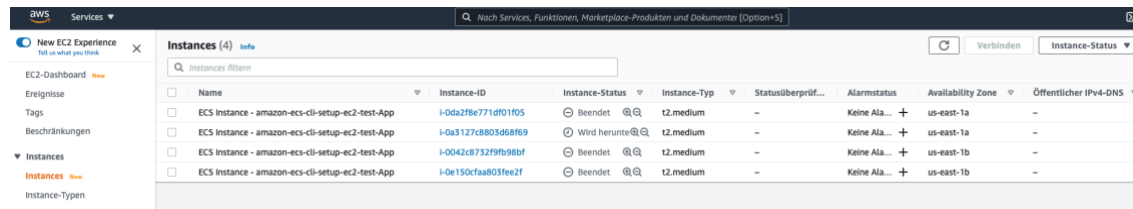
ecs-cli shutting - down!

```
ecs-cli compose down --cluster-config ec2-test-App
```

```
ecs-cli down --force --cluster-config ec2-test-App
```

compose down!

force down!



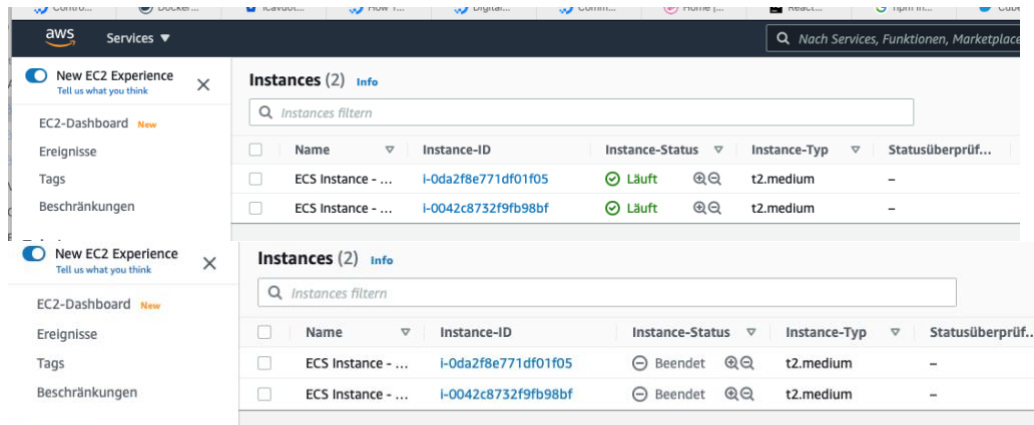
Name	Instance-ID	Instance-Status	Instance-Typ	Statusüberprüf...	Alarmstatus	Availability Zone	Öffentlicher IPv4-DNS
ECS Instance - amazon-ecs-cli-setup-ec2-test-App	i-06a2f8e771d01f05	Beendet	t2.medium	-	Keine Ala...	us-east-1a	-
ECS Instance - amazon-ecs-cli-setup-ec2-test-App	i-0a3127c8803d68f69	Wird herunterg...	t2.medium	-	Keine Ala...	us-east-1a	-
ECS Instance - amazon-ecs-cli-setup-ec2-test-App	i-0042c8732f9f98bf	Beendet	t2.medium	-	Keine Ala...	us-east-1b	-
ECS Instance - amazon-ecs-cli-setup-ec2-test-App	i-0e150cfaa803fee2f	Beendet	t2.medium	-	Keine Ala...	us-east-1b	-

```
ecs-cli down
```

```
$ ecs-cli down
Are you sure you want to delete your cluster? [y/N]
```

stopping all the apps with the down command

y
INFO[0005] Waiting for your cluster resources to be deleted...
INFO[0005] Cloudformation stack status `stackStatus=DELETE_IN_PROGRESS`
INFO[0005] Waiting for your cluster resources to be deleted...
INFO[0005] Cloudformation stack status `stackStatus=DELETE_IN_PROGRESS`
INFO[0068] Cloudformation stack status `stackStatus=DELETE_IN_PROGRESS`
INFO[0130] Deleted cluster `cluster=ec2-test-App`



Name	Instance-ID	Instance-Status	Instance-Typ	Statusüberprüf...
ECS Instance - ...	i-0da2f8e771df01f05	Laufend	t2.medium	-
ECS Instance - ...	i-0042c8732f9fb98bf	Laufend	t2.medium	-

beendet!

if down we start again:

```
ecs-cli up --capability-iam --size 2 --instance-type t2.medium --cluster-config ec2-test-App --ecs-profile ec2-test-App
```

listing the instances/containers

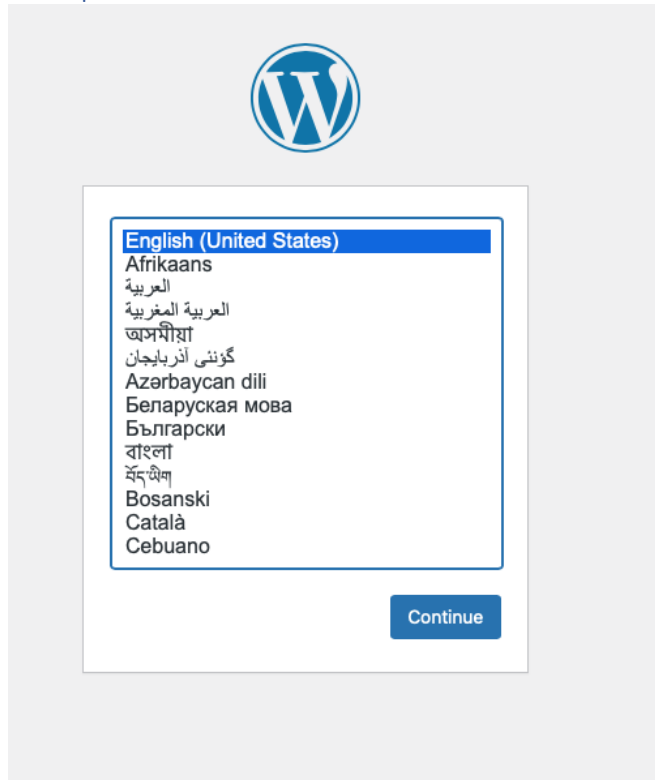
```
ecs-cli ps --cluster-config ec2-test-App
```

listing the instances with ecs-cli ps

```
$ ecs-cli ps --cluster-config ec2-test-App
```

Name	State	Ports	TaskDefinition	Health
ec2-test-App/a8a4e3e913a047989928a5a9aa7eebdd/wordpress	RUNNING	3.92.196.246:80->80/tcp	ec2-project:1	UNKNOWN
ec2-test-App/a8a4e3e913a047989928a5a9aa7eebdd/mysql	RUNNING		ec2-project:1	UNKNOWN

wordpress on 3.92.196.246:80



connection to the MySQL backend works! otherwise you do not get the screen

ecs-cli compose down

```
$ ecs-cli compose down --cluster-config ec2-test-App
```

ecs-cli compose down!

React – App tic tac toe

Image is on docker-hub : `lcavuoti/app-tictactoe:prod-0.1.12`

port: 80:3000

we create a docker-compose file on the local directory, where we start the commands:

```
version: '3'
services:
  tictactoe:
    image: lcavuoti/app-tictactoe:prod-0.1.12
    ports:
      - "80:3000"
```

reload the compose - file

and rerun the compose command:

```
$ ecs-cli compose --project-name ec2-project service up --cluster-config ec2-test-App
```

```
INFO[0001] Using ECS task definition          TaskDefinition="ec2-project:6"
INFO[0001] Updated the ECS service with a new task definition. Old containers will be stopped automatically, and replaced with new ones
desiredCount=1 force-deployment=false service=ec2-project
INFO[0001] Service status                    desiredCount=1 runningCount=1 serviceName=ec2-project
INFO[0011] (service ec2-project) has started 1 tasks: (task 782c905b84a249de9bf857a87b4ef5f3). timestamp="2021-05-16 12:26:17 +0000 UTC"
INFO[0033] Service status                    desiredCount=1 runningCount=2 serviceName=ec2-project
INFO[0102] Service status                    desiredCount=1 runningCount=1 serviceName=ec2-project
INFO[0107] (service ec2-project) has stopped 1 running tasks: (task bfa0c9dd04734cd783992aece7488a64). timestamp="2021-05-16 12:27:51 +0000 UTC"
INFO[0118] ECS Service has reached a stable state desiredCount=1 runningCount=1 serviceName=ec2-project
```

check the instances!

we check the instances :

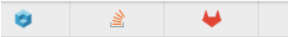
```
$ ecs-cli ps --cluster-config ec2-test-App
```

Name	State	Ports	TaskDefinition	Health
ec2-test-App/782c905b84a249de9bf857a87b4ef5f3/tictactoe	RUNNING	3.88.130.254:80->3000/tcp	ec2-project:6	UNKNOWN
ec2-test-App/bfa0c9dd04734cd783992aece7488a64/tictactoe	STOPPED ExitCode: 0	18.207.153.194:80->80/tcp	ec2-project:5	UNKNOWN

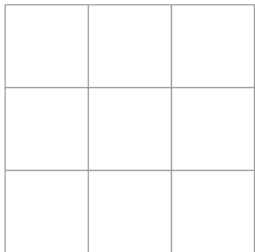
ec2-test-App/fb686493d7b94687a7d1cffe3a159b7d/mysql STOPPED ExitCode: 0
ec2-test-App/fb686493d7b94687a7d1cffe3a159b7d/wordpress STOPPED ExitCode: 0

3.88.130.254:80->80/tcp ec2-project:4 UNKNOWN
ec2-project:4 UNKNOWN

Nicht sicher — 3.88.130.254



Next player: X



tic tac toe runs!

run the browser!

Fully - Managed Services AWS – Fargate

following