# Build Services

Save the following as “Dockerfile” in a directory.

FROM docker.int.thomsonreuters.com:5001/proview/browser/browser-nginx:20170714

Save the following as “nginx.conf” in the same directory as Dockerfile, editing the highlighted portions as needed:

daemon off;

events {

worker\_connections 1024;

}

http {

access\_log /dev/stdout;

# set client body size to 2M #

client\_max\_body\_size 200M;

upstream proview-pfs-branch {

server app:3000;

}

server {

listen 80;

location / {

proxy\_pass http://proview-pfs-branch;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

}

Open Docker console.

Navigate to the folder with Dockerfile and nginx.conf.

Build with the appropriate tag where the Docker image will be stored:

docker build -t docker.int.thomsonreuters.com:5001/proview/proview-pfs-site-site/proview-pfs-branch-nginx:latest.

Push Dockerfile to location (requires credentials):

docker login docker.int.thomsonreuters.com:5001

docker push docker.int.thomsonreuters.com:5001/proview/proview-pfs-site-site/proview-pfs-branch-nginx:latest

Optional logout (best practice):

docker logout docker.int.thomsonreuters.com:5001

# Setup Service in AWS

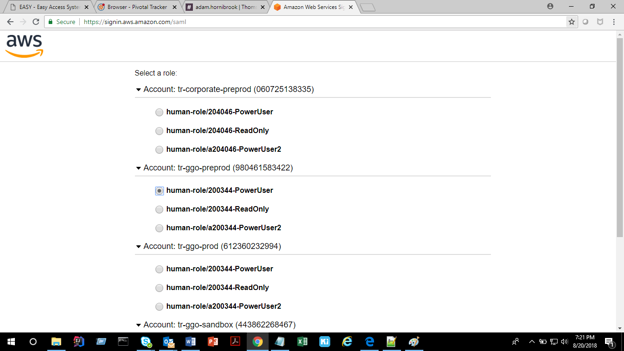
Create the CloudFormation script. Use the .yml file copied in the [Appendix](#Appendix), editing as needed.

Access AWS Login Portal - TR

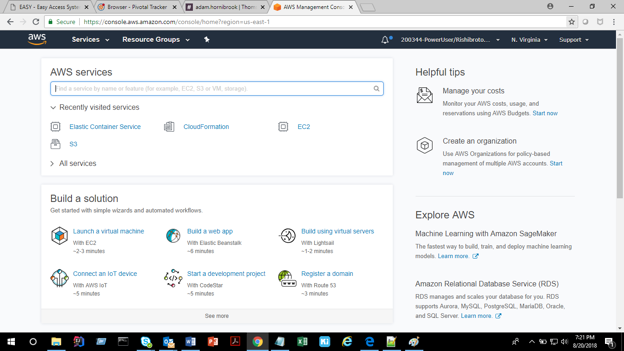
<https://mfs.thomsonreuters.com/adfs/ls/idpinitiatedsignon>

Sign in with credentials

Select ggo preprod



Navigate to the CloudFormation tool.



Upload the .yml file.



Proceed to next and enter in the appropriate stack name matching the prior standard (ex. proview-ci-pfs-branch).



Proceed to Next.

Optionally add tags to the stack (TR standards say they should be, but they aren’t enforced).

Proceed to Create.

Once the service is created, navigate to the Elastic Container Services (ECS) tool in AWS.

Select the desired cluster and open the newly created service by clicking on the hyperlinked name.



Select Create new revision.



Select Configure via JSON at the bottom.



Update the “environment” setting for the new container with the correct environment settings and keys (the initial value should be empty…don’t edit the nginx container environment).

"environment": [

{

"name": "NODE\_ENV",

"value": "ci-branch"

},

{

"name": "ONEPASS\_API\_KEY",

"value": "123ABC"

},

{

"name": "ONEPASS\_API\_KEY\_SECRET",

"value": "123XYZ"

}

],

Click Save.

Click Create.

This will create a new version of the task. Note that it won’t automatically be deployed. Any old tasks must be stopped, a new build/rolling deploy be initiated, or in this setup case, the tasks be set to 1 or more.

Return to the service and choose “Update” (see prior screenshot).

Update “Number of tasks” from 0 to a minimum of 1.

Choose “Next step” for all screens and “Update Service” at the end to save.



# Appendix

CloudFormation – proview-pfs-branch.yml

AWSTemplateFormatVersion: "2010-09-09"

Description: CloudFormation template for the ECS service pfs-branch

Metadata:

"AWS::CloudFormation::Interface":

ParameterGroups:

-

Label:

default: "Load Balancer Configuration"

Parameters:

- loadbalancerName

- commonName

- pathPattern

-

Label:

default: "Tags"

Parameters:

- FinancialIdentifier

- AssetInsightId

- Platform

- Environment

- ResourceOwner

Parameters:

VPC:

Description: The tr\_vpc id for the region

Type: AWS::EC2::VPC::Id

Default: vpc-9c891dfa

TaskDefinitionName:

Description: Service task definition name. Should be {ProviewEnvironment}-servicename

Type: String

Default: proview-ci-pfs-branch

TargetGroupName:

Description: Service target group name. Should be {ProviewEnvironment}-servicename

Type: String

Default: proview-ci-pfs-branch

commonName:

Description: Optional (Do not specify if pathPattern); the host-header DNS name; proviewsites.ci.int.thomsonreuters.com; proviewsites.qa.int.thomsonreuters.com; proviewsites.thomsonreuters.com

Type: String

Default: pfs-branch.proviewsites.ci.int.thomsonreuters.com

pathPattern:

Description: Optional (Do not specify if commonName); the path-pattern to use by the load balancer to forward request to the service

Type: String

loadbalancerName:

Description: The name of the load balancer which will host the common name

Type: String

Default: proview-ci-proviewsites-alb

ECSCluster:

Description: The name of the ECS cluster to put the service

Type: String

Default: proview-ci

ClusterScriptName:

Description: Cloud formation script name used to create ECS cluster

Type: String

Default: proview-ci

LogGroup:

Description: Cloud watch log group

Type: String

Default: proview-ci

DesiredCount:

Description: The number of identity processes to run

Type: String

Default: 0

FinancialIdentifier:

Description: "Cost Center ID"

Type: String

Default: 704615010

AssetInsightId:

Description: Asset Insight ID

Type: String

Default: 200344

Platform:

Description: Platform tag

Type: String

Default: Proview

Environment:

Description: Environment tag

Type: String

Default: DEVELOPMENT

AllowedValues:

- DEVELOPMENT

- QUALITY ASSURANCE

- PRODUCTION

ResourceOwner:

Description: Owner tag

Type: String

Default: Thomson-MobileDev-Server@thomsonreuters.com

Conditions:

CreatePathPattern: !Not [!Equals [!Ref pathPattern,""]]

InProdEnvironment: !Equals [!Ref Environment,"PRODUCTION"]

Resources:

TargetGroup:

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

Name: !Ref TargetGroupName

Port: 80

Protocol: HTTP

VpcId: !Ref VPC

HealthCheckPath: "/v1/statuscheck"

HealthCheckProtocol: HTTP

HealthCheckTimeoutSeconds: 10

HealthCheckIntervalSeconds: 30

HealthyThresholdCount: 3

Tags:

-

Key: platform

Value: !Ref Platform

-

Key: tr:financial-identifier

Value: !Ref FinancialIdentifier

-

Key: tr:application-asset-insight-id

Value: !Ref AssetInsightId

-

Key: tr:environment-type

Value: !Ref Environment

-

Key: Name

Value: proview-build

-

Key: tr:resource-owner

Value: !Ref ResourceOwner

TaskDef:

Type: AWS::ECS::TaskDefinition

Properties:

ContainerDefinitions:

-

Name: nginx

Cpu: 512

MemoryReservation: 512

PortMappings:

-

ContainerPort: 80

Protocol: tcp

Privileged: true

Essential: true

Image: "docker.int.thomsonreuters.com:5001/proview/proview-pfs-site-site/proview-pfs-branch-nginx:latest"

LogConfiguration:

LogDriver: awslogs

Options:

awslogs-group: !Ref LogGroup

awslogs-region: us-east-1

awslogs-stream-prefix: !Join ["-",[!Ref TaskDefinitionName,"nginx"]]

Environment:

-

Name: environment

Value: !Ref Environment

Links:

- app

-

Name: app

Cpu: 512

MemoryReservation: 512

PortMappings:

-

ContainerPort: 3000

Protocol: tcp

Privileged: true

Essential: true

Image: "docker.int.thomsonreuters.com:5001/proview/proview-pfs-site-site/proview-pfs-branch-app:20180730.1"

LogConfiguration:

LogDriver: awslogs

Options:

awslogs-group: !Ref LogGroup

awslogs-region: us-east-1

awslogs-stream-prefix: !Ref TaskDefinitionName

Family: !Ref TaskDefinitionName

NetworkMode: bridge

TaskRoleArn:

"Fn::ImportValue": !Sub "${ClusterScriptName}-AMIContainerTaskRole"

loadbalancerInfo:

Type: "Custom::loadbalancerInfo"

Properties:

ServiceToken: !If [InProdEnvironment,"arn:aws:lambda:us-east-1:612360232994:function:Proview-CloudFormation-LB-Rule-Lookup","arn:aws:lambda:us-east-1:980461583422:function:Proview-CloudFormation-LB-Rule-Lookup"]

loadbalancerName:

!Ref loadbalancerName

InternalPublicListener:

Type: "AWS::ElasticLoadBalancingV2::ListenerRule"

DependsOn:

- loadbalancerInfo

Properties:

Actions:

-

TargetGroupArn: !Ref TargetGroup

Type: forward

Conditions:

-

Field: !If [CreatePathPattern,path-pattern,host-header]

Values:

- !If [CreatePathPattern,!Ref pathPattern,!Ref commonName]

ListenerArn: !GetAtt loadbalancerInfo.ListenerArn

Priority: !GetAtt loadbalancerInfo.Priority

serviceDef:

Type: AWS::ECS::Service

DependsOn:

- InternalPublicListener

Properties:

Role:

"Fn::ImportValue": !Sub "${ClusterScriptName}-AMIServiceRole"

TaskDefinition:

!Ref TaskDef

DesiredCount: 0

LoadBalancers:

-

TargetGroupArn:

!Ref TargetGroup

ContainerPort: 80

ContainerName: nginx

Cluster:

!Ref ECSCluster

ServiceName: pfs-branch