*1.​* What are the states of a Promise?

A promise can be one of 3 states.

* Pending: initial state, neither fulfilled nor rejected.
* Fulfilled: meaning that the operation was completed successfully.
* Rejected: meaning that the operation failed.

2. What is the difference between Promise.all and Promise.allSettled?

- Promise.all will reject as soon as one of the Promise in the array rejects.

- Promise.allSettled will never reject, it will resolve once all Promises in the array have

either rejected or resolved.

3.Given the following code, implement functions “yesterday” and “lastWeek” that would return corresponding timestamps.

const today = () => ( new Date( new Date().setHours( 0, 0, 0, 0 ) ) ).getTime();

const dayBefore = timestamp => timestamp - dayInMiliseconds();

const dayInMiliseconds = () => 24 \* 3600 \* 1000;

const yesterday = dayBefore(today());

const lastWeek = today() - 7 \* dayInMiliseconds()

4. Implement a simple Express app that would just serve the contents of the “files” directory statically on port 8080.

// server.js

const express = require('express');

// Define Express App

const app = express();

const PORT = process.env.PORT || 8080;

// Serve Static Assets

app.use(express.static('public'));

app.listen(PORT, () => {

    console.log('Server connected at:', PORT);

});

5. Given the following DynamoDB table querying code, implement proxying functions named “asc” and “desc” that would accept params object and pass it on to query in a way that enforces ordering the returned data in ascending and descending order.

const query = async params => new Promise(( resolve, reject ) => {

    client.query( {... params,

        TableName: TABLE,

        }, ( err, rows ) => {

        if ( err ) {

            return reject( err );

        }

        resolve(( rows || {} ).Items || [] );

    });

});

const asc = () => {

    let params = {

        TableName: 'Events',

        KeyConditionExpression: 'Organizer = :organizer',

        Limit: 1,

        ScanIndexForward: true,    // true = ascending, false = descending

        ExpressionAttributeValues: {

            ':organizer': organizer

        }

    };

    return query(params)

}

const desc = () => {

    let params = {

        TableName: 'Events',

        KeyConditionExpression: 'Organizer = :organizer',

        Limit: 1,

        ScanIndexForward: false,    // true = ascending, false = descending

        ExpressionAttributeValues: {

            ':organizer': organizer

        }

    };

    return query(params)

}

6. In the following example, a lambda function is subscribed to a CloudWatch stream. Implement the getData function that would fit the example, which returns a list of recorded events.

exports.handler = async (event) => {

    try {

        const payload = Buffer.from( event.awslogs.data, 'base64' );

        const res = await getData( payload );

        if ( res ) {

            // ...

            // Do something with data

            let eventlist = [];

            for (const logevent of logevents) {

                let log = JSON.parse(logevent.message);

                eventlist.push(log);

            }

            return eventlist;

        }

    }

    catch ( e ) {

        console.error( e );

    }

};

7.What is the difference between Query and Scan operations in DynamoDB?

In terms of performance, I think it's good practice to design your table for applications to use Query instead of Scan. Because a scan operation always scan the entire table before it filters out the desired values, which means it takes more time and space to process data operations such as read, write and delete.

8. What are projection expressions in DynamoDB?

To read data from a table, you use operations such as GetItem, Query, or Scan.

A projection expression is a string that identifies the attributes that you want.

9. How would you make items in a DynamoDB table expire after a period of time?

You can make Amazon DynamoDB table entries expire automatically by setting a time-to-live timestamp attribute. This timestamp is in Unix Epoch time format which is number of seconds that have elapsed since 1 January 1970 00:00:00.

10. How would we map ports in docker using cli command?

Docker run port mapping

$ docker run -p 127.0.0.1:80:8080/tcp ubuntu bash

11. What is the difference between docker stop and docker kill commands?

docker stop is Stop a running container (send SIGTERM, and then SIGKILL after grace period) The main process inside the container will receive SIGTERM, and after a grace period, SIGKILL.

docker kill is Kill a running container (send SIGKILL, or specified signal) The main process inside the container will be sent SIGKILL, or any signal specified with option --signal.

12. Given the numbers in the following tuple sequence, how would we obtain a list of their squares?

numbers = (1, 2, 3, 4, 5)

sqared\_numbers = [number \*\* 2 for number in numbers]

13. Define a function is\_palindrome that would return True if the input string is palindrome (a word, number, phrase, or other sequence of characters which reads the same backward as forward - e.g. madam, racecar), False otherwise:

def isPalindrome(input\_string):

    result = input\_string == input\_string[::-1]

    return result

14. Given the following data structure, print a list of pages that have GET as their type and 403 as their status:

list\_string = [

    {

        "type": "GET",

        "status": 200,

        "page": "example.com/one"

    },

    {

        "type": "POST",

        "status": 200,

        "page": "example.com/two"

    },

    {

        "type": "GET",

        "status": 404,

        "page": "example.com/three"

    },

    {

        "type": "POST",

        "status": 403,

        "page": "example.com/four"

    },

    {

        "type": "GET",

        "status": 500,

        "page": "example.com/five"

    },

    {

        "type": "GET",

        "status": 403,

        "page": "example.com/six"

    },

    {

        "type": "POST",

        "status": 403,

        "page": "example.com/seven"

    },

    {

        "type": "GET",

        "status": 403,

        "page": "example.com/eight"

    }

]

def getItem(item\_list):

    result = []

    for item in item\_list:

        if item['type'] == "GET" and item['status'] == 403:

            result.append(item)

    print(result)

getItem(list\_string)

15. Assume​ ​$books​ is an array of books with the following format:

$tbData = array

(

    array

        (

            "title" => "title1",

            "purchased\_at" => "2020-05-13 09:49:54"

        ),

    array

        (

           "title" => "title2",

           "purchased\_at" => "2020-07-13 09:49:54"

        ),

    array

        (

            "title" => "title3",

            "purchased\_at" => "2020-09-13 09:49:54"

        ),

    array

        (

            "title" => "title4",

            "purchased\_at" => "2020-05-19 09:49:54"

        )

);

usort($tbData, function ($a, $b) {

     return ($a["purchased\_at"]) < ($b["purchased\_at"]);

});

print\_r($tbData);

16. The following code creates a file in the submitted path. What is wrong with it and how would you improve it?

<?php

$directory = isset($\_POST['directory']) ? $\_POST['directory'] : '/root\_dir';

$subdirectory = isset($\_POST['subdirectory']) ? $\_POST['subdirectory'] : '/main\_subdir';

$path = rtrim($directory, '/') . '/' . trim($subdirectory, '/');

$new\_file = $path . '/' . 'new\_file.txt';

// Create path if it doesn't exist.

if(!is\_dir($path)) {

    mkdir($path, 0777, true);

}

// Now create the file.

$fp=fopen($new\_file,"w");

17. The following code returns the names of products from the local db. How can it be improved?

<?php

$ids = get\_ids();

$result = $connection->query("SELECT `product\_name` FROM `products` WHERE `product\_id` IN ('$ids')");

$products = $result->fetch\_row();

return $products;

18. [Here's a template we ruined earlier.](https://drive.google.com/file/d/18D9oUuUHZFol_AEspbUdPvMEl4aNlcXf/view?usp=sharing) ​[H](https://drive.google.com/file/d/18D9oUuUHZFol_AEspbUdPvMEl4aNlcXf/view?usp=sharing)ow would you fix and improve it? Go ahead, make it a masterpiece!

I’ve tried to fix it using python yaml module.

import yaml

document = open('ecsInfrastructureLB.yml').read()

INPUT = yaml.load(document, Loader=yaml.BaseLoader)

with open(r'ecsInfrastructureLB.yml', 'w') as file:

    documents = yaml.dump(INPUT, file)

This is perfect result. ☺

AWSTemplateFormatVersion: '2010-09-09'

Description: CloudFormation template for an ECS AutoScaling|LB setup - Matomo WPMU

  DEV.

Parameters:

  CertificateArn1:

    Default: ''

    Type: String

  ClusterSize:

    Default: ''

    Description: How many ECS hosts do you want to initially deploy?

    Type: Number

  EC2SubnetPrivate:

    Default: ''

    Description: The subnets to place EC2 instances in.

    Type: AWS::EC2::Subnet::Id

  EC2SubnetPrivate2:

    Default: ''

    Description: The subnets to place EC2 instances in.

    Type: AWS::EC2::Subnet::Id

  EC2SubnetPrivate3:

    Default: ''

    Description: The subnets to place EC2 instances in.

    Type: AWS::EC2::Subnet::Id

  EC2SubnetPublic:

    Default: ''

    Description: The subnets to place EC2 instances in.

    Type: AWS::EC2::Subnet::Id

  EC2SubnetPublic2:

    Default: ''

    Description: The subnets to place EC2 instances in.

    Type: AWS::EC2::Subnet::Id

  EC2SubnetPublic3:

    Default: ''

    Description: The subnets to place EC2 instances in.

    Type: AWS::EC2::Subnet::Id

  InstanceType:

    AllowedValues:

    - ''

    ConstraintDescription: must be a valid EC2 instance type.

    Default: ''

    Description: WebServer EC2 instance type

    Type: String

  KeyName:

    ConstraintDescription: must be the name of an existing EC2 KeyPair.

    Default: keyname

    Description: The EC2 Key Pair to allow SSH access to the instances

    Type: AWS::EC2::KeyPair::KeyName

  OperatorEMail:

    AllowedPattern: ([a-zA-Z0-9\_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)

    ConstraintDescription: must be a valid email address.

    Default: email

    Description: EMail address to notify if there are any scaling operations

    Type: String

  VPCID:

    Default: ''

    Description: VPC ID

    Type: String

  VPCsecgroup:

    ConstraintDescription: must be a valid security group

    Default: ''

    Description: The VPC security group to use for the instances

    Type: String

Resources:

  AutoscalingRole:

    Properties:

      AssumeRolePolicyDocument:

        Statement:

        - Action:

          - sts:AssumeRole

          Effect: Allow

          Principal:

            Service:

            - application-autoscaling.amazonaws.com

      Path: /

      Policies:

      - PolicyDocument:

          Statement:

          - Action:

            - application-autoscaling:\*

            - cloudwatch:DescribeAlarms

            - cloudwatch:PutMetricAlarm

            - ecs:DescribeServices

            - ecs:UpdateService

            - cloudformation:SignalResource

            Effect: Allow

            Resource: '\*'

        PolicyName: service-autoscaling

    Type: AWS::IAM::Role

  CPUAlarmHigh:

    Properties:

      AlarmActions:

      - Ref: WebServerScaleUpPolicy

      AlarmDescription: Scale-up if CPU > 75% for 5 minutes

      AlarmName: matomo-wpmudev-cpu

      ComparisonOperator: GreaterThanOrEqualToThreshold

      Dimensions:

      - Name: AutoScalingGroupName

        Value:

          Ref: ECSAutoScalingGroup

      EvaluationPeriods: '1'

      MetricName: ''

      Namespace: AWS/EC2

      Period: '300'

      Statistic: Average

      Threshold: '75'

      Unit: Percent

    Type: AWS::CloudWatch::Alarm

  CPUAlarmLow:

    Properties:

      AlarmActions:

      - Ref: WebServerScaleDownPolicy

      AlarmDescription: Scale-down if CPU < 40% for 60 minutes

      AlarmName: matomo-wpmudev-cpu-down

      ComparisonOperator: LessThanThreshold

      Dimensions:

      - Name: AutoScalingGroupName

        Value:

          Ref: ECSAutoScalingGroup

      EvaluationPeriods: '1'

      MetricName: ''

      Namespace: AWS/EC2

      Period: '3600'

      Statistic: Average

      Threshold: '40'

      Unit: Percent

    Type: AWS::CloudWatch::Alarm

  ECSAutoScalingGroup:

    CreationPolicy:

      ResourceSignal:

        Count: '1'

        Timeout: PT60M

    Properties:

      AutoScalingGroupName: matomo-wpmudev

      DesiredCapacity: '2'

      HealthCheckGracePeriod: '180'

      HealthCheckType: ELB

      LaunchConfigurationName:

        Ref: ECSLaunchConfig

      MaxSize: '6'

      MinSize: '2'

      TargetGroupARNs:

      - ECSTargetGroup

      TerminationPolicies:

      - ClosestToNextInstanceHour

      VPCZoneIdentifier:

      - Ref: EC2SubnetPublic

      - Ref: EC2SubnetPublic2

      - Ref: EC2SubnetPublic3

    Type: AWS::AutoScaling::AutoScalingGroup

    UpdatePolicy:

      AutoScalingRollingUpdate:

        MaxBatchSize: '1'

        MinInstancesInService: '0'

        PauseTime: PT5M

        SuspendProcesses:

        - HealthCheck

        - ReplaceUnhealthy

        - AZRebalance

        - AlarmNotification

        - ScheduledActions

  ECSCluster:

    Properties:

      ClusterName: matomo-wpmudev

    Type: AWS::ECS::Cluster

  ECSLaunchConfig:

    Metadata:

      AWS::CloudFormation::Init:

        config:

          files:

            /tmp/matomo-test.txt:

              content: This is a test.

              context:

                name: matomo-test.txt

      Comment": EC2 Instance take off

    Properties:

      BlockDeviceMappings:

      - DeviceName: /dev/xvda

        Ebs:

          DeleteOnTermination: 'true'

          VolumeSize: '40'

          VolumeType: gp2

      - DeviceName: /dev/xvdcz

        Ebs:

          DeleteOnTermination: 'true'

          VolumeSize: '22'

          VolumeType: gp2

      IamInstanceProfile: ecsInstanceRole

      ImageId: ami-0cca5d0eeadc8c3c4

      InstanceType:

        Ref: InstanceType

      KeyName:

        Ref: KeyName

      SecurityGroups:

      - Ref: VPCsecgroup

      UserData:

        Fn::Base64: 'Content-Type: multipart/mixed; boundary="==BOUNDARY=="

          MIME-Version: 1.0

          --==BOUNDARY==

          MIME-Version: 1.0

          Content-Type: text/x-shellscript; charset="us-ascii"

          #!/bin/bash

          echo "deploy60" > /etc/deploy-new-docker.txt

          export REGION="us-east-2"

          export ECS\_CLUSTER=${ECSCluster}

          yum update -y -q

          yum install -y -q aws-cli aws-cfn-bootstrap mysql

          echo ''DOCKER\_STORAGE\_OPTIONS="--storage-driver overlay2"'' > /etc/sysconfig/docker-storage

          service docker restart

          aws ecr get-login --no-include-email --region us-east-2 > login.sh

          chmod +x login.sh

          ./login.sh

          echo ECS\_CLUSTER=$ECS\_CLUSTER >> /etc/ecs/ecs.config

          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource ECSLaunchConfig

          --region ${AWS::Region}

          /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource ECSAutoScalingGroup

          --region ${AWS::Region}

          --==BOUNDARY==--

          '

    Type: AWS::AutoScaling::LaunchConfiguration

  ECSListener443:

    Properties:

      DefaultActions:

      - TargetGroupArn: ECSTargetGroup

        Type: forward

      LoadBalancerArn: ECSLoadBalancerV2

      Port: '443'

      Protocol: HTTPS

    Type: AWS::ElasticLoadBalancingV2::Listener

  ECSLoadBalancerV2:

    Properties:

      LoadBalancerAttributes:

      - Key: deletion\_protection.enabled

        Value: 'true'

      Name: matomo-wpmudev

      Scheme: internet-facing

      Subnets:

      - Ref: EC2SubnetPublic

      - Ref: EC2SubnetPublic2

      - Ref: EC2SubnetPublic3

      Type: application

    Type: AWS::ElasticLoadBalancingV2::LoadBalancer

  ECSServiceRole:

    Properties:

      AssumeRolePolicyDocument:

        Statement:

        - Action:

          - sts:AssumeRole

          Effect: Allow

          Principal:

            Service:

            - ecs.amazonaws.com

      Path: /

      Policies:

      - PolicyDocument:

          Statement:

          - Action:

            - elasticloadbalancing:DeregisterInstancesFromLoadBalancer

            - elasticloadbalancing:DeregisterTargets

            - elasticloadbalancing:Describe\*

            - elasticloadbalancing:RegisterInstancesWithLoadBalancer

            - elasticloadbalancing:RegisterTargets

            - ec2:Describe\*

            - ec2:AuthorizeSecurityGroupIngress

            - ec2:disassociateAddress

            - ecs:StartTask

            - cloudformation:SignalResource

            Effect: Allow

            Resource: '\*'

        PolicyName: ecs-service

    Type: AWS::IAM::Role

  ECSTargetGroup:

    Properties:

      HealthCheckIntervalSeconds: '60'

      HealthCheckPath: /index.php

      HealthCheckPort: '443'

      HealthCheckProtocol: HTTPS

      HealthCheckTimeoutSeconds: '20'

      HealthyThresholdCount: '2'

      Name: matomo-wpmudev

      Port: '443'

      Protocol: HTTPS

      Tags:

      - Key: Name

        Value: matomo-wpmudev

      - Key: Service

        Value: WPMUDEV

      TargetGroupAttributes:

      - Key: deregistration\_delay.timeout\_seconds

        Value: '30'

      - Key: slow\_start.duration\_seconds

        Value: '300'

      UnhealthyThresholdCount: '2'

      VpcId: VPCID

    Type: AWS::ElasticLoadBalancingV2::TargetGroup

  EcsSecurityGroup:

    Properties:

      GroupDescription: Matomo WPMU DEV ECS security group

      GroupName: matomo-wpmudev-ecs

      Tags:

      - Key: Service

        Value: WPMUDEV

      - Key: Name

        Value: matomo-wpmudev-ecs

      VpcId: VPCID

    Type: AWS::EC2::SecurityGroup

  WebServerScaleDownPolicy:

    Properties:

      AdjustmentType: ChangeInCapacity

      AutoScalingGroupName:

        Ref: ECSAutoScalingGroup

      Cooldown: '60'

      ScalingAdjustment: '-1'

    Type: AWS::AutoScaling::ScalingPolicy

  WebServerScaleUpPolicy:

    Properties:

      AdjustmentType: ChangeInCapacity

      AutoScalingGroupName:

        Ref: ECSAutoScalingGroup

      Cooldown: '300'

      ScalingAdjustment: '1'

    Type: AWS::AutoScaling::ScalingPolicy

  service:

    DependsOn: ECSListener443

    Properties:

      Cluster: ECSCluster

      DesiredCount: '2'

      LoadBalancers:

      - ContainerName: matomo-wpmudev

        ContainerPort: '443'

        TargetGroupArn: ECSTargetGroup

      PlacementConstraints:

      - Type: distinctInstance

      PlacementStrategies:

      - Field: host

        Type: spread

      Role: ECSServiceRole

      TaskDefinition: taskdefinition

    Type: AWS::ECS::Service

  taskdefinition:

    Properties:

      ContainerDefinitions:

      - Essential: 'true'

        Hostname: ''

        Image: ''

        LinuxParameters:

          InitProcessEnabled: 'false'

        Memory: '3584'

        Name: matomo-wpmudev

        PortMappings:

        - ContainerPort: '443'

      Family: matomo-wpmudev

      NetworkMode: host

    Type: AWS::ECS::TaskDefinition