

$$\{ \quad \quad \quad \}$$

# { AWS Shared Responsibility Model }

/01	IAM Users/Groups	< Create Users/Groups, Key-id & key-value, Assign group features. Create Policies/Roles, Attach/detach, JSON/Visual editor.
/02	IAM Policies/Roles	
/03	CLI config	< Command line configuration, commands.
/04	Cloudtrail	< Events history, Trails in S3 bucket.
/05	Boto3	< Install Boto3 in Flask, Migrate data events.

}

# /01 IAM Users/Groups

III

## IAM = Identity and Access Management, Global service

- Root account created by default, shouldn't be used or shared
- Users are people within your organization and can be grouped.
- Groups only contain users, not other groups



## IAM = Current HawkinCloud User & Groups

- 1 • The following access key indicates which user is actively using the configuration in the CLI.
- 2 • We created an Admin group and assigned each group member a user in the group policy.
- 3 • It displays the group member "Aliyev" and the policies that are attached to them, creating events in the project based on those policies.



Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Group Admin
AWSCloudTrail_ReadOnlyAccess	AWS managed	Directly
IAMReadOnlyAccess	AWS managed	Directly
KMS	Customer managed	Directly
KMS1	Customer managed	Directly
KMS3	Customer managed	Directly
KMSEnableKeyRotation	Customer managed	Directly
KMSReadPolicy	Customer managed	Directly
ListUsers	Customer managed	Directly

< Aliyev Omar >

## /02 IAM Policies/Roles

IV

### 1 IAM = Custom JSON & AdministratorAccess Policies

- 2 • Users or Groups can be assigned JSON or given by AWS existing policy documents called policies
- 3 • These policies define the permissions of the users

```
4 {  
5   "Version": "2012-10-17",  
6   "Statement": [  
7     {  
8       "Effect": "Allow",  
9       "Action": "*",  
10      "Resource": "*"   
11    }  
12  ]  
13 }
```

The policy provided is a very permissive policy that allows access to all actions on all resources. This policy essentially grants full access to all AWS resources and actions. It is important to note that this policy should be used with caution and only granted to trusted users who require such broad access.

Job Function

```
8 {  
9   "Version": "2012-10-17",  
10  "Statement": [  
11    {  
12      "Effect": "Allow",  
13      "Action": [  
14        "iam:GenerateCredentialReport",  
15        "iam:GenerateServiceLastAccessedDetails",  
16        "iam:Get*",  
17        "iam:List*",  
18        "iam:SimulateCustomPolicy",  
19        "iam:SimulatePrincipalPolicy"  
20      ],  
21      "Resource": "*"   
22    }  
23  ]  
24 }
```

This policy is more restrictive than the previous one you provided. It grants specific permissions related to the AWS Identity and Access Management (IAM) service. Overall, this policy grants the user permission to view and generate reports about IAM resources, list and get information about IAM resources, and simulate the effects of IAM policies on specified resources. This policy is still quite permissive, but it is more targeted and appropriate for users who require IAM-specific permissions.

Non - Job Function

< Aliyev Omar >

## /03 CLI config

V

### AWS ACCESS = Management Console, CLI, SDK

- AWS Management Console (protected by password + MFA)
- **AWS Command Line Interface (CLI): protected by access keys**
- **AWS Software Developer Kit (SDK) - for code: protected by access keys**

### AWS ACCESS = CLI

- Access Keys are generated through the AWS Console
- Users manage their own access keys
- Access Key ID ~= username
- Secret Access Key ~= password

```
aliyevom@Aliyev-2 ~ % aws configure
AWS Access Key ID [*****ZPNU]:
AWS Secret Access Key [*****9fZZ]:
Default region name [us-east-1]:
Default output format [None]:
aliyevom@Aliyev-2 ~ %
```

### AWS ACCESS = CLI include

- Create and manage EC2 instances, load balancers, and other resources
- Create and manage IAM users, groups, roles, and policies
- Create and manage S3 buckets and objects
- Manage your VPC and network resources
- Configure and manage your AWS CLI settings

```
aliyevom@Aliyev-2 ~ % aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "aliyev",
      "UserId": "AIDAJ54TXXXXXX",
      "Arn": "arn:aws:iam::123456789012:user/aliyev",
      "CreateDate": "2023-02-26T22:00:59+00:00"
    }
  ]
}
```

# /04 Cloudtrail

VI

## AWS Clodutrail = Provides governance, compliance and audit for your AWS Account

- CloudTrail is enabled by default!
- Get an history of events / API calls made within your AWS Account by:
  - Console
  - SDK
  - CLI
  - AWS Services
- Can put logs from CloudTrail into CloudWatch Logs or S3
- A trail can be applied to All Regions (default) or a single Region.
- If a resource is deleted in AWS, investigate CloudTrail first!



## AWS Clodutrail = HawkinCloud S3 bucket, History of events, IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudtrail:Get*",
        "cloudtrail:Describe*",
        "cloudtrail:List*",
        "cloudtrail:LookupEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

S3 bucket

aws-cloudtrail-logs-783210027625-3ca5ae9f

Event name	Event time	User name	Event source	Resource type	Resource name
ConsoleLogin	March 16, 2023, 18:11:16 UTC...	root	signin.amazonaws.com	-	-
StopInstances	March 16, 2023, 03:24:21 UTC...	root	ec2.amazonaws.com	AWS::EC2::Instance, A...	i-0e478fa24eeff3137, i-09375970ba0b6040a, i-0d7e503ea4e1e19a5
RunInstances	March 16, 2023, 03:23:27 UTC...	root	ec2.amazonaws.com	AWS::EC2::VPC, AWS::E...	vpc-05ca7a5826a1c308a, ami-0bd9d09e7460b08a8, eni-0c44a281304ed5250, i-09375970ba0b6040a
RunInstances	March 16, 2023, 03:09:48 UTC...	root	ec2.amazonaws.com	AWS::EC2::AMI, AWS::E...	ami-0bd9d09e7460b08a8, First, sg-06c803296979ed298
CreateImage	March 16, 2023, 03:08:25 UTC...	root	ec2.amazonaws.com	AWS::EC2::AMI, AWS::E...	ami-0bd9d09e7460b08a8, i-0e478fa24eeff3137
AuthorizeSecurityGre...	March 16, 2023, 02:53:26 UTC...	root	ec2.amazonaws.com	AWS::EC2::SecurityGroup	sg-0c470e725db16c1c
AuthorizeSecurityGre...	March 16, 2023, 02:49:10 UTC...	root	ec2.amazonaws.com	AWS::EC2::SecurityGroup	sg-0c470e725db16c1c
SharedSnapshotWala...	March 16, 2023, 02:42:24 UTC...	-	ec2.amazonaws.com	-	-
RunInstances	March 16, 2023, 02:42:22 UTC...	root	ec2.amazonaws.com	AWS::EC2::VPC, AWS::E...	vpc-05ca7a5826a1c308a, ami-005f9685cb30f234b, eni-07505462228462753, i-0e478fa24eeff313...
DeleteUser	March 16, 2023, 02:21:13 UTC...	root	iam.amazonaws.com	AWS::IAM::User	Michael
RemoveUserFromGro...	March 16, 2023, 02:21:13 UTC...	root	iam.amazonaws.com	AWS::IAM::User, AWS::I...	Michael, Admin
DeleteLoginProfile	March 16, 2023, 02:21:12 UTC...	root	iam.amazonaws.com	AWS::IAM::User	Michael
AssociateInstance...	March 16, 2023, 02:20:14 UTC...	root	ec2.amazonaws.com	AWS::EC2::Instance	i-0d7e503ea4e1e19a5
AddRoleToInstanceP...	March 16, 2023, 02:19:45 UTC...	root	iam.amazonaws.com	AWS::IAM::InstancePro...	Ec2-Config, Ec2-Config
CreateInstanceProfile	March 16, 2023, 02:19:45 UTC...	root	iam.amazonaws.com	AWS::IAM::InstancePro...	Ec2-Config, amawsiam:783210027625instance-profile/Ec2-Config, APASMWXUTAUUNE4SKQCS
AttachRolePolicy	March 16, 2023, 02:19:44 UTC...	root	iam.amazonaws.com	AWS::IAM::Policy, AWS::...	amawsiam:aws-policy/IAMReadOnlyAccess, Ec2-Config
CreateRole	March 16, 2023, 02:19:44 UTC...	root	iam.amazonaws.com	AWS::IAM::Role, AWS::I...	amawsiam:783210027625role/Ec2-Config, AGDA1MWXUTAUJULSH5SMH2, Ec2-Config
AttachUserPolicy	March 16, 2023, 02:07:34 UTC...	root	iam.amazonaws.com	AWS::IAM::User, AWS::I...	aliyev, amawsiam:aws-policy/IAMReadOnlyAccess
StartInstances	March 16, 2023, 02:03:52 UTC...	root	ec2.amazonaws.com	AWS::EC2::Instance	i-0d7e503ea4e1e19a5
DisableKey	March 16, 2023, 01:28:36 UTC...	root	kmk.amazonaws.com	AWS::KMS::Key, AWS::K...	7d96c5d7-289e-4b0c-bc5a-6d62162e796, amawsiam:us-east-1:783210027625key/7d96c5d7-28...
StopInstances	March 15, 2023, 19:36:36 UTC...	root	ec2.amazonaws.com	AWS::EC2::Instance	i-0d7e503ea4e1e19a5
ConsoleLogin	March 15, 2023, 19:26:17 UTC...	root	signin.amazonaws.com	-	-

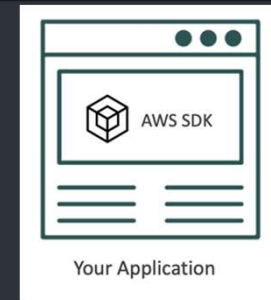
< Aliyev Omar >

## /05 Boto3

VII

### AWS Software Development Kit = AWS SDK

- Enables you to access and manage AWS services programmatically
- Supports:
  - SDKs (JavaScript, Python, PHP, .NET, Ruby, Java, Go, Node.js, C++)
  - Mobile SDKs (Android, iOS, ...)
  - IoT Device SDKs (Embedded C, Arduino, ...)



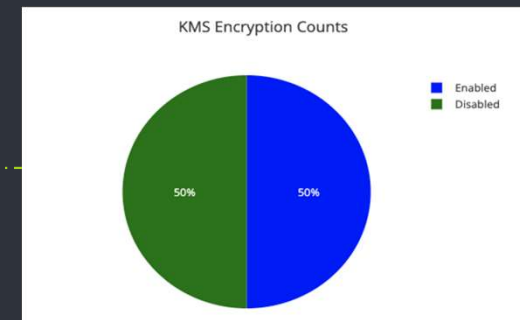
### AWS CLI is built on AWS SDK for Python = HawkinCloud

```
client = boto3.client('cloudtrail')
# List all events for a specific trail
# List all trails
response = client.lookup_events()
print(response)
events=[]
for event in response['Events']:
    events.append({
        'event_id': event['EventId'],
        'event_name': event['EventName'],
        'event_time': event['EventTime'],
        'event_detail': event['CloudTrailEvent'],
    })
print("Event: ", event['EventId'])
# Render the template with the events data
print(events)
return render_template('dashboard.html', name=current_user.username, events=events)
```

```
@app.route('/kms_data')
def kms_data():
    keys = ['Enabled', 'Disabled']
    enabled_count = 0
    disabled_count = 0
    client = boto3.client('kms')
    keys_list = client.list_keys()
    for key in keys_list['Keys']:
        rotation_response = client.get_key_rotation_status(
            KeyId=key['KeyId'])
        if(rotation_response["KeyRotationEnabled"] is True):
            enabled_count+=1
        else:
            disabled_count+=1
    rotation_count = [enabled_count, disabled_count]

    return jsonify({
        'keys': keys,
        'rotation_count': rotation_count,
    })
```

Event ID	Event Name	Event Time	Event Detail
7882edc7-4c47-4b5a-bd5e-4b1cbe994d7	DescribeEventAggregates	2023-03-16 20:14:51:04:00	[{"eventVersion": "1.08", "userIdentity": {"type": "Root", "principalId": "783210027625", "arn": "arn:aws:iam::783210027625:root", "accountId": "783210027625", "accessKeyId": "ASIA3MWXUTJUQNU52P3E", "sessionContext": {"sessionIssuer": {"webIdFederationData": {}}, "attributes": {"creationDate": "2023-03-16T22:11:16Z", "mfaAuthenticated": "false"}}, "eventTime": "2023-03-17T00:14:51Z", "eventSource": "health.amazonaws.com", "eventName": "DescribeEventAggregates", "awsRegion": "us-east-1", "sourceIPAddress": "66.31.54.133", "userAgent": "AWS Internal", "requestParameters": {"aggregateField": "eventSource", "filter": {"eventStatusCodes": ["open", "upcoming"], "startTimes": [{"from": "Mar 10, 2023 12:14:51 AM"}]}, "responseElements": null, "requestId": "7683068a-6dbf-4653-99ad-b0e3eb689911", "eventId": "7882edc7-4c47-4b5a-bd5e-4b1cbe994d7", "readOnly": true, "eventType": "AwsApiCall", "managementEvent": true, "recipientAccountId": "783210027625", "eventCategory": "Management", "sessionCredentialFromCor



< Aliyev Omar >

## **HawkinCloud Status**

Since the most recent day of class-presentation day, our group has made substantial efforts in working on both the frontend and backend.

With almost 2 months into the initiation of the AWS project, HawkinCloud, we have made drastic changes on the front-end, with the creation displaying significant metrics as they relate to the AWS Cloudtrail as displayed by dashboard.html-integrated with Flask.

We continue to meet more than once a week, either virtually or in person. Significant progress on both the frontend and backend have fast-tracked HawkinCloud.



## **HawkInCloud Timeline (Recap)**

Phase I - Conceptualization

As discussed in our introductory presentation

Phase II - Initialization

Frontend (HTML), Backend (Python), and Integration (Flask) are connected and arranged for primary functionality

Phase III - Optimization

CloudTrail/Boto3 are implemented for HawkInCloud's full AWS services

End.html

X

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

THANK  
YOU



}

<http://hawkincloud.github.io/dev/>