Login Details for experiementing account

Email: space.in.the.news@gmail.com
Passsword: e[G‘uaX?9i”aieJ

0.1 Users

Username: Ben
Access Type: AWS Managament Console access
Console Password: sdjf248r+(23ef)-af2+
Sign in link: https://266867367924.signin.aws.amazon.com/console

1 Accroynyms

- CIDR = Classless Interdomain Routing.
- VPC = Virual Private Cloud.
- NACLs = Network Access Control Lists.
- AMI = Amazon Machine Image.
- EBS = Elastic Block Store.
- RDP = Remote Desktop Protocol.

2 Basics and Introduction

2.1 AWS Global Infrastructure

- Availability zone = one or more data centres
- Each region consists of two or more Availability zones.
- Each region is completely independent.
- Every region is connected via a high bandwidth, fully redundant network. (does not use the internet - more consistent result than internent)

2.2 AWS Pricing

2.2.1 Compute

Amounnt of resources (eg. CPU, RAM, . . .)

2.2.2 Storage

Quantity of data (size allocated for data normally)

2.2.3 Outbound Data Transfer

Don’t pay for inbound data, but do pay for the quantity of data that is transferred out from all services.

3 Create Billing Alarm

1. Navigate to **Billing & Cost Management Dashboard** from the main page (can be searched for (search for 'billing')).
2. select **preferences** on the left

Can also (from the main page (services drop down)) go to CloudWatch.

1. Go to **CloudWatch**
2. Change region to N. Virginia
3. select **In Alarm**
4. create alarm
5. billing alarm
6. total
7. leave settings the same
8. click next to set up SNS **Simple Notification Service**
9. Select **In Alarm**
10. Create new topic
11. Set topic and email
12. set up topic
13. confirm email subscription
14. can be viewed in the SNS Console
15. Click next
16. set alarm name
17. next
18. click **create alarm**
19. done

4 AWS Identity and Access Management Service (IAM)

Can apply an **IAM Policy** to an **IAM User** or **IAM Group** (to define permissions)
IAM User = an entity that represents a person or service.
IAM Group = a collection of users and have poilicies attached to them.
IAM Role = Roles are "assumed" by trusted entities and can be used for delegation.
Access Key = Consists of an Access key ID and a secret access key. Used for programmatic access to the API.

5 Create IAM User and Group

1. Select **Services** drop down in the main bar
2. select **IAM** under the **Security, Identity, & Compliance** section.
3.

5.1 Create a User Group

1. click on **User Groups** (or the number below it)
2. Click **Create Group**
   (blue button, top right)
3. Set Group name
4. Attach Permissions Policy.
   for full administrator access, select **AdministratorAccess**
5. Click **Create Group**
   (blue button, bottom right)

5.1.1 Create User (to be added to the user group)

1. Under **Access management** on the LHS panel, select **Users**.
2. Click **Add User** (blue button, top right).
3. Type in a user name
4. select access type and provide a password to be used by user if necessary.
5. click next
   ... onto permisions
6. Add User to the existing **Admins** group.
7. Once confirmed, can sign in using the link provided (using the username and password just created)

6 Amazon Virtual Private Cloud (VPC)

VPC = a logically isolated portion of the AWS cloud within a region.
Subnets are created within availability zones (AZs).
- Can have Public and Private subnets
- Public subnets means instances running in it get a public IP address and can access the internet gateway directly.
- Private subnets cannot access the internet gateway directly. (there is a way to get around this though)
Route Table is used to configure the VPC router.

Route Tables can be assigned to the subnets.
Can launch virtual servers into VPC subnets.
Each VPC has a different block of IP addresses.
CIDR = Classless Interdomain Routing. (IP addressing concept)
Can Create multiple VPCs within each region.
Each subnet has a block of IP addresses from the CIDR block.

To view VPCs, go to Services, VPC (under networking and content delivery)

7 Security Groups & Network Access Control Lists (NACLs)

- NACLs = Type of Firewalls

  NACLs Apply at the subnet level.

  Stateless

  processes rules in order

  Supports allow and deny rules

- Security Groups apply at the instance level

  applies to instance level communication (within security group)

  can be applied to instances in any subnet

  stateful

  evaluates all rules

  Supports allow rules only

7.1 How to configure Security Groups and NACLs

7.1.1 Network ACLs

The last rule (inbound and outbound) is a deny so that if an allow rule is not found, this drops the traffic.
To add or edit a rule click **Edit inbound rules** (top right of bottom pannel).

7.2 Stateful vs Stateless Firewalls

Stateful will allow the return traffic automatically.
Stateless will check for an allow rule for both connections (in and out).

8 AWS Public and Private Services

9 Installation of AWS Command Line Interface

Google for **aws cli install** and install version 2.

10 Section 2: Amazon Elastic Compute Cloud (EC2)

- Public IP address

  Lost when the instance is stopped

  Associated with a private IP address on the instance

- Private IP address

  Retained when the instance is stopped

  Used in public and private Subnets

- Elastic IP address

  Static Public IP address

  Are charged if not used

  Associated with a private IP address on the instance

  Can be moved between instances and Elastic Network Adapters

NAT = Network Adress Translation.
- Translates private to elastic IP (via a NAT Gateway).
- Receives traffic from private IP address and forwards it on using an Elastic-IP address to the internet
- has to be running in a public subnet (not private).

10.1 Launching an Amazon EC2 Instance

Amazon Machine Image (AMI)
- comes with an operating system and any preconfigured software
- and EBS (Elastic Block Store) snapshot
port 22 = secure shell
for windows - port 3389 = RDP (Remote Desktop Protocol)

10.2 Connecting to Amazon EC2 Instance

To update all packages installed on the instance can run:

```
sudo yum update -y
```

10.3 Create Website Using User Data

10.4 Using Key Pairs with Amazon EC2

11 Docker with AWS

https://docs.aws.amazon.com/AmazonECS/latest/userguide/docker-basics.html

```
ben@Bens-MacBook-Pro-2 keys % aws ecr create-repository --repository-name
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-1:266867367924:repository/hello-repo
```

```
        "registryId": "266867367924",
        "repositoryName": "hello-repository",
        "repositoryUri": "266867367924.dkr.ecr.us-east-1.amazonaws.com/hello-repos
        "createdAt": "2021-12-04T21:54:12+00:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
```

Push your image to Amazon Elastic Container Registry

```
aws ecr create-repository --repository-name [hello-repository] --region [r
---
output:
---
{
    "repository": {
        "registryId": "aws_account_id",
        "repositoryName": "hello-repository",
        "repositoryArn": "arn:aws:ecr:region:aws_account_id:repository/hel
        "createdAt": 1505337806.0,
        "repositoryUri": "aws_account_id.dkr.ecr.region.amazonaws.com/hell
    }
}
---
docker tag hello-world [aws_account_id].dkr.ecr.[region].amazonaws.com/[he
---
aws ecr get-login-password | docker login --username AWS --password-stdin
---
docker push [aws_account_id].dkr.ecr.[region].amazonaws.com/[hello-reposit
--- clean up ---
aws ecr delete-repository --repository-name [hello-repository] --region [r
```

Used in the above:

```
aws_account_id (registryId) = 266867367924
region = us-east-1
hello-repository (repositoryName) = hello-repository
```

aws ecr get-login-password — docker login –username AWS –password-stdin 266867367924.dkr.ecr.us-east-1.amazonaws.com

docker tag hello-world 266867367924.dkr.ecr.us-east-1.amazonaws.com/hello-repository

docker push 266867367924.dkr.ecr.us-east-1.amazonaws.com/hello-repository