CloudShell Demo

1. Log into the AWS Management Console using credentials associated with a root user account.

   Ensure you have taken the time to memorise these credentials.

2. Click on the CloudShell icon in the bar at the top of the console.

3. Demonstrate making the font larger on the CLoudShell terminal. Do this by clicking on the cog icon.

4. Press enter, without typing anything, to move through the four prompts (Access Key, Secret Access Key, Region, Input mode).

5. Demonstrate the `--version` command and the `configure` command.

6. Demonstrate complete freedom to perform commands. You are logged in as the root user, so you have wide permissions.

   a. `aws s3 ls`
   b. `aws iam list-users`

7. Now create an IAM user using this command:

   ```
   aws iam create-user --user-name <name>
   ```

8. Now give that IAM user an access key, using this command:

```
aws iam create-access-key –user-name <name>
```

   Copy the access key ID and secret access key into Notepad.

   Explain that you have now given this IAM user the means to programmatically access AWS resources.

9. Run the `aws configure` command.
   You will now be prompted to input
   Input the Access Key ID and Secret Access key for the IAM user you just created.
   Feel free to log into an unusual Region, and select and unusual input mode (e.g. JSON).

10. Demonstrate that this IAM user has permission to do very little.
    For example, you could try:

```
aws s3 ls
aws iam list-users
aws ec2 describe-instances
```

11. So, now what you're going to demonstrate is the use of the root user account to add permissions to the IAM user you just created.

```
aws iam attach-user-policy
     --user-name <user-name>
     --policy-arn
arn:aws:iam::aws:policy/AmazonS3FullAccess
```

12. Now, log in as the IAM user and demonstrate that you can use:

```
aws s3 ls
```

Discussion point: Think about the command `aws ec2 run-instances`. What are all the things that need to be in place already, if an EC2 instance is to be launched?

13. Try another command as the IAM user:

```
aws ec2 describe-instances

aws ec2 describe-vpcs

aws ec2 create-security-group --group-name <name>
--description <description> --vpc-id <vpc-id>
```

14. Now, let's suppose you want to allow your user to be able to describe vpcs.

You are going to create a custom policy.

(i) We will achieve this with two commands. First, we are going to run `aws iam create-policy`. Then, we are going to run `aws iam attach-user-policy`.

(ii) First, you need to write a JSON policy and store it in S3 (to have a policy stored somewhere is a prerequisite for the `aws iam create-policy` command).

```
echo '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeVpcs",
      "Resource": "*"
    }
  ]
}' | aws s3 cp - s3://your-bucket-name/iam-
policy.json
```

(iii) Run the `create-policy` command:

```
aws iam create-policy \
    --policy-name DescribeVPCsPolicy \
    --policy-document file://describe-vpcs-policy.json
```

(iv) Now run the `attach-user-policy` command:

```
aws iam attach-user-policy \
    --user-name <username> \
    --policy-arn
arn:aws:iam::aws:policy/DescribeVPCsPolicy
```

Hints and Tips

1. You need to have two dashes before "version". However, "configure" does not have any dashes.

2. If AWS CloudShell asks you about logs, try pressing the `q` key on the keyboard.