Day 1 - Create Key Pair

The Nautilus DevOps team is strategizing the migration of a portion of their infrastructure to the AWS cloud. Recognizing the scale of this undertaking, they have opted to approach the migration in incremental steps rather than as a single massive transition. To achieve this, they have segmented large tasks into smaller, more manageable units. This granular approach enables the team to execute the migration in gradual phases, ensuring smoother implementation and minimizing disruption to ongoing operations. By breaking down the migration into smaller tasks, the Nautilus DevOps team can systematically progress through each stage, allowing for better control, risk mitigation, and optimization of resources throughout the migration process.

For this task, create a key pair with the following requirements:

- Name of the `key pair` should be `nautilus-kp`.
- Key pair `type` must be `rsa`

What is an AWS Key Pair?

An AWS key pair is a security credential consisting of a public and private key used for secure SSH authentication to EC2 instances. AWS stores the public key, while you download and securely store the private key. This cryptographic pair ensures that only authorized users with the private key can access the instances.

Why Key Pairs Matter for Migration

As the Nautilus DevOps team migrates to AWS, secure access to EC2 instances is fundamental.

Key pairs provide cryptographic authentication that's more secure than passwords and enables automated, scalable infrastructure management.

Solution

We run these commands in the console to generate our key pair

```
aws ec2 create-key-pair \
    --key-name nautilus-kp \
    --key-type rsa \
    --query 'KeyMaterial' \
    --output text > nautilus-kp.pem
```

This command creates a new RSA key pair named `nautilus-kp` in AWS, extracts only the private key material from the response, and saves it as a text file named `nautilus-kp.pem` in your current directory. The public key is automatically stored in AWS while you receive and must securely store the private key locally.

```
~ on  (us-east-1) ➜ aws --version
aws-cli/1.40.19 Python/3.10.17 Linux/5.15.0-1083-gcp botocore/1.38.20

~ on  (us-east-1) ➜ aws ec2 create-key-pair \
    --key-name nautilus-kp \
    --key-type rsa \
    --query 'KeyMaterial' \
    --output text > nautilus-kp.pem

~ on  (us-east-1) ➜ ls
nautilus-kp.pem

~ on  (us-east-1) ➜
```

Next we run this command

```
aws ec2 describe-key-pairs --key-names nautilus-kp
```

This command queries AWS to retrieve and display detailed information about the `nautilus-kp` key pair, confirming it was created successfully and showing its properties like key ID, fingerprint, and type. This verification step ensures the key pair exists in AWS before attempting to use it with EC2 instances.

```
~ on  (us-east-1) ➜ aws ec2 describe-key-pairs --key-names nautilus-kp
{
    "KeyPairs": [
        {
            "KeyPairId": "key-0a071b7abd52eac2d",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2026-01-05T18:23:07.092Z",
            "KeyName": "nautilus-kp",
            "KeyFingerprint": "bb:41:ea:78:63:87:8a:d4:fe:30:9d:6a:ca:bf:ab:d0:30:53:6e:f4"
        }
    ]
}
```

After we have to set the correct permissions to the key:

```
chmod 400 nautilus-kp.pem
```

This sets the file to read-only for you (the owner) and removes all access for everyone else, which is a security requirement for SSH to accept the key.

```
~ on  (us-east-1) ➜ chmod 400 nautilus-kp.pem

~ on  (us-east-1) ➜ ls -la nautilus-kp.pem
-r-------- 1 root root 1675 Jan  5 18:23 nautilus-kp.pem
```

Security Best Practices Implemented

- ✅ Used RSA-2048 encryption (industry standard)
- ✅ Set restrictive file permissions (400)
- ✅ Private key stored locally, never shared
- ✅ Verified key creation before proceeding

Additional Recommendations

- Store backup in encrypted password manager
- Rotate keys every 90 days
- Never commit .pem files to version control
- Add *.pem to .gitignore