

Lab 1 – EC2

Task – Create an EC2 and connect to it

- 1) Login to your AWS IAM Account and access the **EC2** portal
- 2) From the portal click on the **Launch instance** button and follow the wizard
- 3) Ensure you are downloading a **new key pair** or using an **existing key pair** you have access to for an **SSH** connection
- 4) Connect to your EC2 via **SSH command, SSH client or EC2 Instance Connect** (via AWS GUI)

```
ssh -i <private_key> <user>@<public_ip>
```

Stretch Goal

Create a second EC2 and attempt to connect to it from the first EC2. As you will be connected to the first EC2, you can only connect via an SSH command, and you have to consider the key pair transfer.

Lab 2 – S3 Buckets

Task – Create an S3 Bucket to host a static website

- 1) Login to your AWS IAM account and access the **S3** portal
- 2) From the portal click on the **Create Bucket** button and follow the wizard to create a unique bucket in a region of your choice. Uncheck 'Block all public access'
- 3) Clone down this repo <https://gitlab.com/Reece-Elder/devops-m5-s3> and make some changes to the html, js and css files
- 4) Using the S3 GUI upload your updated files to your bucket
- 5) Within the bucket change the properties of the bucket so 'Static Website Hosting' which is near the bottom

- 6) Change the individual properties of the files in the bucket to allow public reading
- 7) Access the Static Website Hosting endpoint to see your website

Lab 3 – RDS – MySQL

Task 1 – Creating an RDS

Using the RDS portal in AWS, create a 'Free Tier' MySQL instance with default options making note of the username and password.

Task 2 – Accessing your RDS

Modify the Security group of your RDS adding the inbound rule to allow SSH from this security group, make note of the ID of this SG. Create a new EC2 that is using this SG as well as the same VPC and subnet.

- 1) Connect to this EC2 via SSH, EC2 instance connect or any other method
- 2) Update the package manager and install mysql client using apt
- 3) Login to your database using ``mysql -h <rds endpoint> -P 3306 <username> -p`
- 4) Enter your password in the next line

Using example from previous Linux modules, clone down this repo

<https://gitlab.com/Reece-Elder/devops-m5-rds> in your instance and add the sql file to your database. (Tip, use the input output Linux commands)

Grab the following info from the database, no need to save the data but keep note of the commands needed to get the data:

```
All movies with a runtime over 100
All movies with more than 1000 votes
All movies in order of popularity
All people with an id between 10000 - 12000
All people in name ascending order (limit to 1000)
```

Lab 4 – ELB

Task – Create a load balancer between two instances

Prerequisites:

2x t2.micro EC2 in the same SG with ports 80 and 22 open. Ensure EC2s are spread across 2x subnets.

- 1) Connect to both instances and install nginx on each
- 2) Modify the default html by going to `/var/www.html`. For example make one ec2 print out A and the other B so you can tell the difference
- 3) On the AWS EC2 portal go to Elastic Load Balancer and follow the wizard to set up an ELB. Its important to ensure the target group you create contains your EC2 instances
- 4) Access the endpoint of your ELB and refresh the page to see the load balancer working

Stretch goal – Add a third EC2 to your ELB. This one should have nginx installed and show a different HTML file.

Remember to delete your load balancer after creating it!

Lab 5 – VPC

Task 1 – Create a VPC from scratch

Create a VPC with a CIDR block of 10.10.0.0/20 that has 2x public subnets connected. Your VPC must contain the following:

- VPC
- CIDR Blocks
- 2x Public Subnets
- Network gateway
- Route Tables
- Routes
- Security Group



Follow the process logically and attempt to create this. Once you have created your VPC test it works by creating an EC2 connected to a subnet in your VPC and try to SSH into it.

Its important you make this MANUALLY and not using the automated wizard for your VPC.

Task 2 – Create a VPC using the wizard

Create a VPC with a CIDR block of 10.20.0.0/16 with 2x public subnets and 2x private subnets with no NAT. You should use the VPC wizard to create this through the VPC portal.

Create an EC2 and assign it a subnet in the new VPC and test you can connect to it.