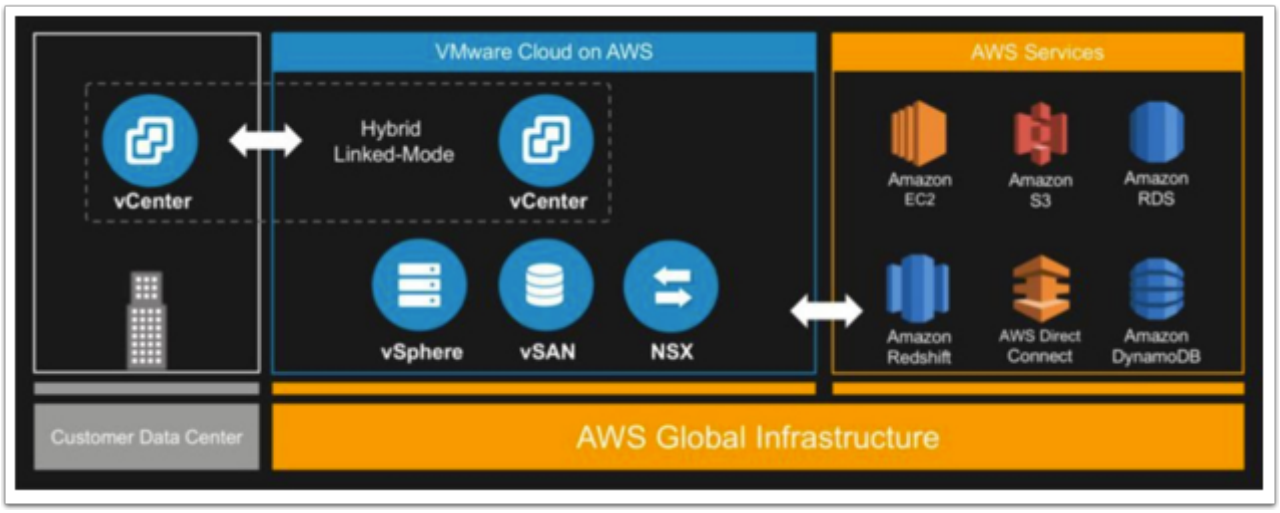
Lab 03 - SDDC Networking & Native AWS Integration

Introduction

One of the most compelling reasons to adopt VMware Cloud on AWS is to integrate your existing systems which sit in your VMware Cloud environment, with application platforms that reside in your AWS Virtual Private Cloud (VPC) environment. The integration which VMware and AWS have created allows for these services to communicate, for free, across a private network address space for services such as EC2 instances, which connect into subnets within a native AWS VPC, or with platform services that have the ability to connect to a VPC Endpoint, such as S3 Storage.

Understanding Integration with AWS Services



As the above diagram illustrates, the VMware stack not only sits next to the AWS services but is tightly integrated with these services. This introduces a new way of thinking about how to design and leverage AWS services with your VMware SDDC. Some integrations our customers are using include:

- VMware front-end and RDS backend
- VMware back-end and EC2 front-end

- AWS Application Load Balancer (ELBv2) with VMware front-end (pointing to private IPs)
- Lambda, Simple Queueing Service (SQS), Simple Notification Service (SNS), S3, Route53, and Cognito
- AWS Lex, and Alexa with the VMware Cloud APIs

These are only a few of the integrations we've seen. Many different services that can be integrated into your environment. In this exercise, we'll be exploring integrations with both AWS Simple Storage Service (S3) and AWS Relational Database Service (RDS).

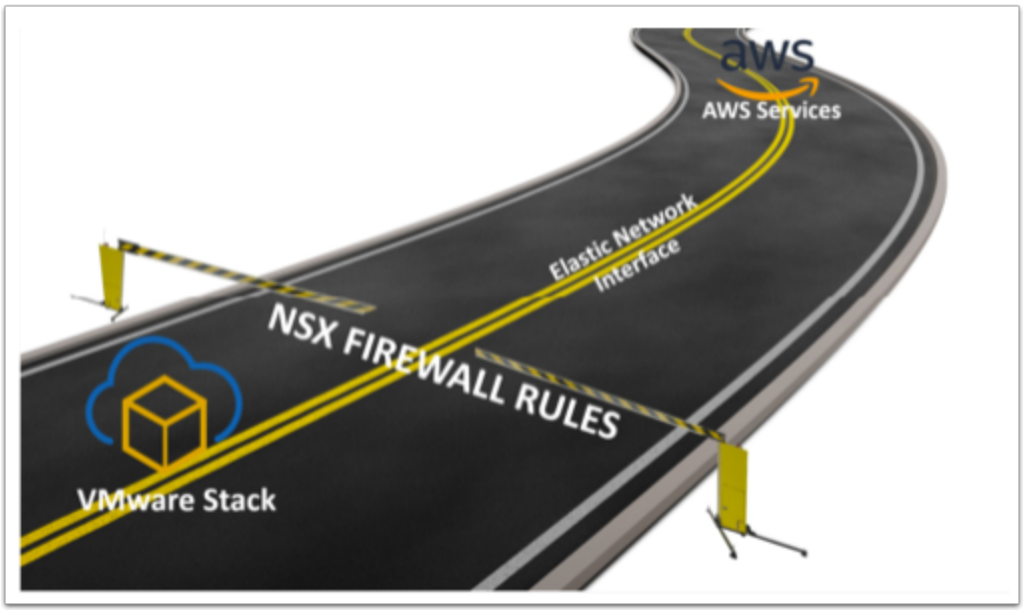
How are these integrations possible?

In addition to sitting within the AWS Infrastructure, there is an Elastic Network Interface (ENI) connecting VMware Cloud on AWS and the customer's Virtual Private Cloud (VPC), providing a high-bandwidth, low-latency connection between the VPC and the SDDC. This is where the traffic flows between the two technologies (VMware and AWS). To leverage native AWS services on your SDDCs, deploy your AWS EC2 workloads in the same availability zone to avoid cross-AZ traffic charges.

How is traffic across the ENI secured?

From the VMware side (see image below), the ENI comes into the SDDC at the Compute Gateway (NSX Edge). This means, on this end of the technology we allow and disallow traffic from the ENI with NSX Firewall rules. By default, no ENI traffic can enter the SDDC. Think of this as a security gate blocking traffic to and from AWS Services on the ENI until the rules are modified.

On the AWS Services side, Security Groups are utilized. For those who are not familiar with Security Groups, they act as a virtual firewall for different services (VPCs, Databases, EC2 Instances, etc). This should be configured to deny traffic to and from the VMware SDDC unless otherwise configured.



Note: There is a requirement in this lab to have completed all the steps in **Lab 2 Working with your SDDC.**

TASKS

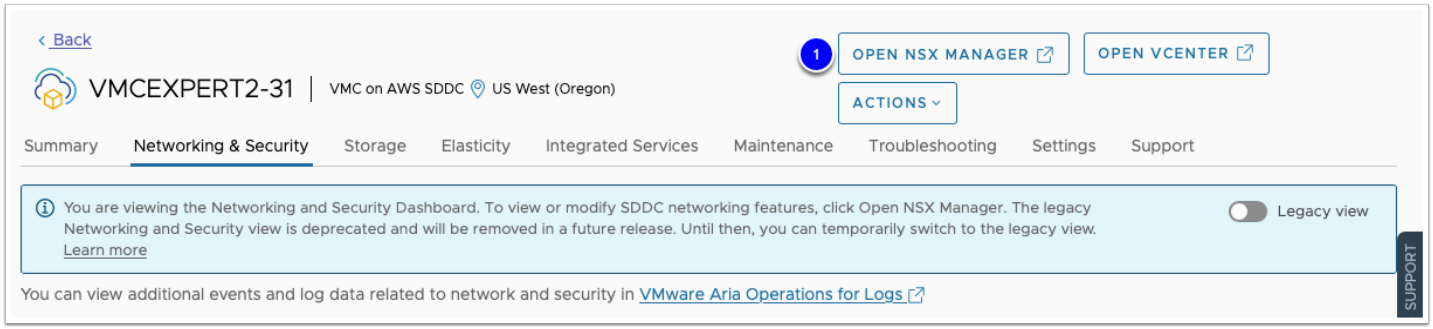
In this lab, you will configure service integration and consumption between the SDDC and AWS Connected VPC. We will use the web server VMs you created in the previous lab to consume services in AWS. We start by consuming an RDS database. We then have optional exercises where you'll consume other services such as ELB, & NFS

In this exercise, everything has been configured on the AWS side for you. You will however walk through how to open AWS traffic to come in and out of your VMware Cloud on AWS SDDC.

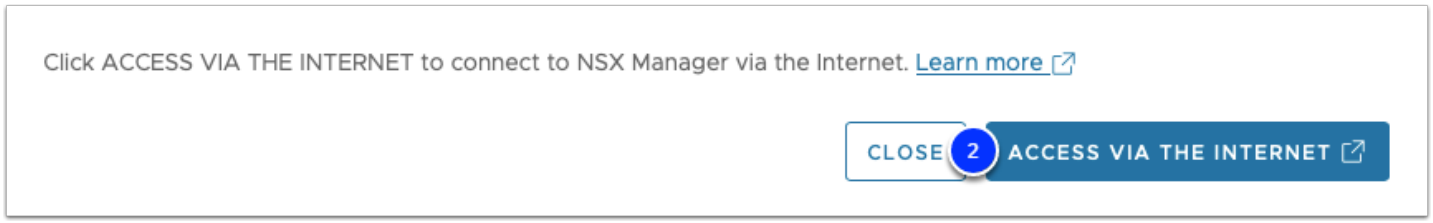
Task 1 - Create Security Groups

Before we can communicate between our SDDC and the connected VPC, we need to allow traffic through the ENI. We will start by create a security group we will use in the firewall rules to allow traffic to and from the AWS RDS.

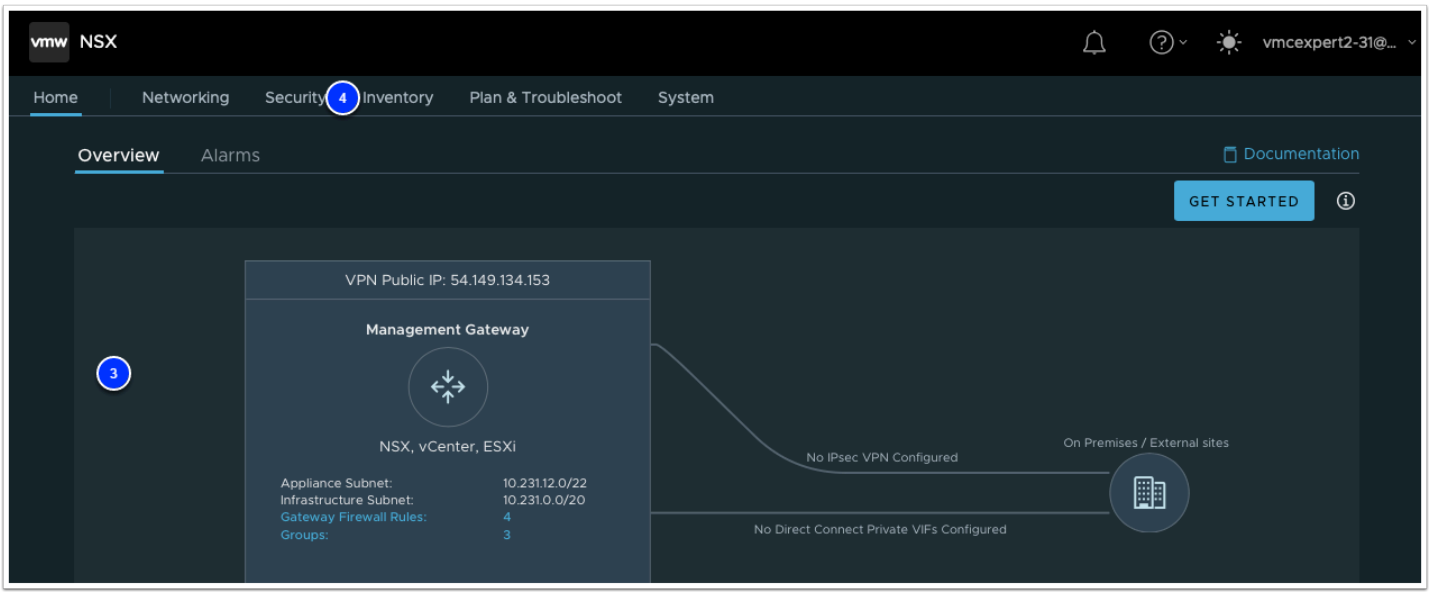
1. In the VMware Cloud on AWS portal click the **OPEN NSX MANAGER** button



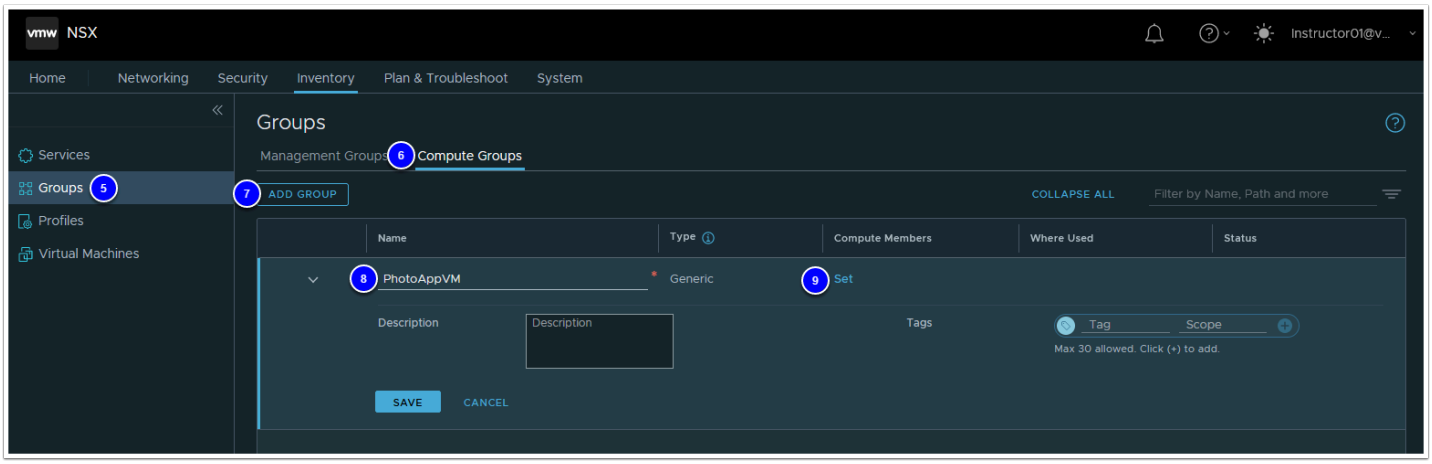
2. Click **ACCESS VIA THE INTERNET** to connect to NSX Manager UI



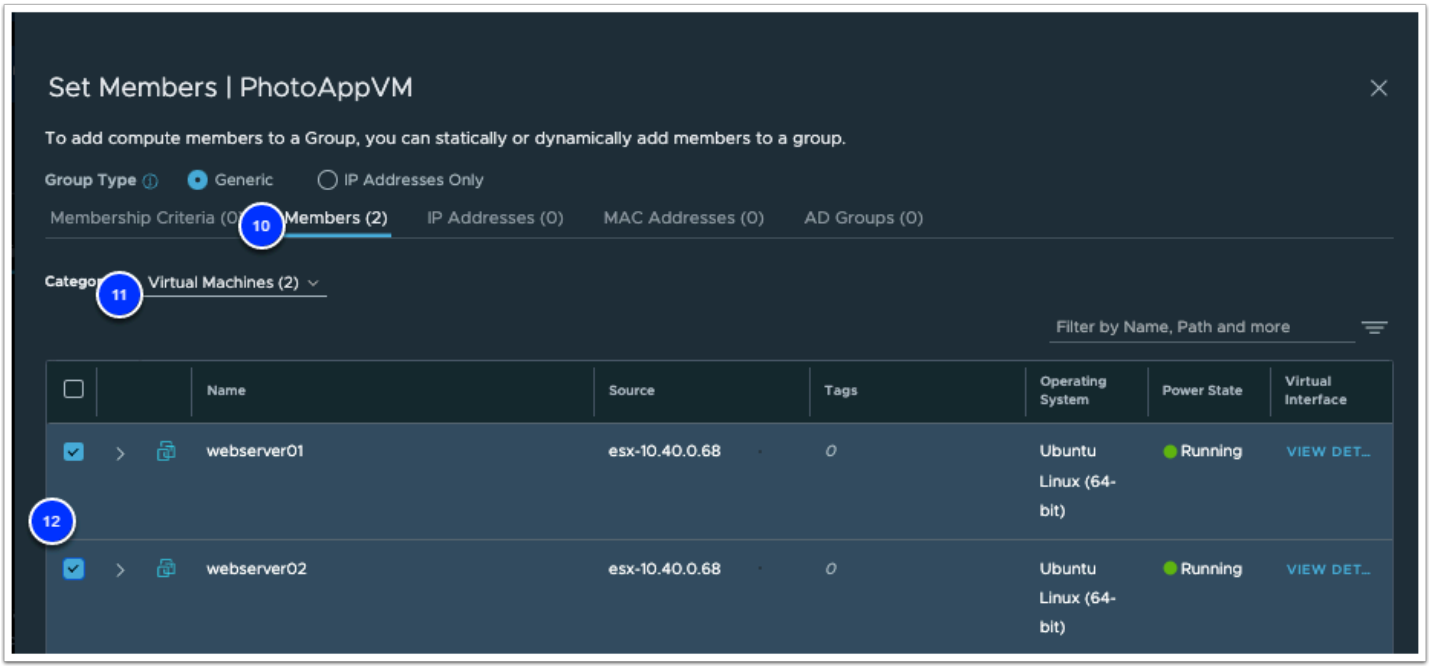
3. Wait till page with NSX Manager will be loaded and you will see Overview dashboard.
4. Click on **Inventory** tab



5. Click **Groups** in the left pane
6. Click **Compute Groups**
7. Click **ADD GROUP**
8. Name: **PhotoAppVM**
9. Click the **Set** link



10. In the popup, Select **Members** Tab
11. From the Drop Down change the Category from **Groups** to **Virtual Machines**
12. Check the box next to **webserver01** and **webserver02**
13. Click **Apply**
14. Click **Save**



Task 2 - Create Gateway Firewall rule

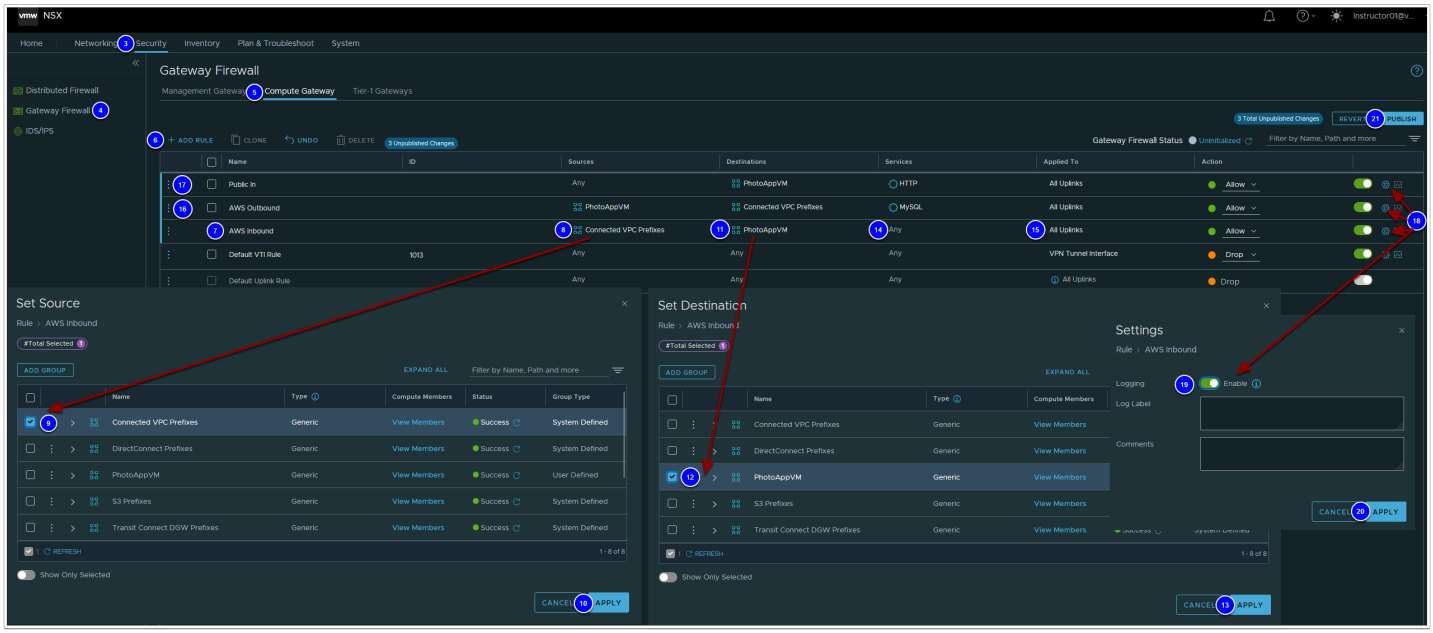
We will now create the required firewall rules to allow the PhotoAppVM access to Services running in the Connected VPC and vice versa.

1. If your NSX Manager UI tab is active then go to step #3. If you already closed NSX Manager tab then Select your SDDC, if you aren't currently within it, then click **View Details**
2. Click the **OPEN NSX MANAGER** button and click **ACCESS VIA THE INTERNET** to connect to NSX Manager UI. Wait till page with NSX Manager will be loaded and you will see **Home - Overview** dashboard.
3. Click **Security** tab in your NSX Manager UI
4. Click **Gateway Firewall** in the left pane
5. Click and select **Compute Gateway**
6. Click **+ADD RULE**
7. Click on the "**New Rule**" Text and enter **AWS Inbound**
8. Hover over the Source field and click on the blue **Edit Pencil**
9. In the popup, Select **Connected VPC Prefixes**
10. Click **Apply**
11. Hover over the **Destination** field and click on the blue **Edit icon**
12. In the popup, Select **PhotoAppVM**
13. Click **Apply**
14. Leave Service as **Any**
15. Leave applied to as **All Uplinks**
16. REPEAT STEPS: 4 - 13 to create an additional rule with the following changes:
    - Name: **AWS Outbound**

- Source: **PhotoAppVM**
- Destination: **Connected VPC Prefixes**
- Service: **MySQL**

17. REPEAT STEPS: 4 - 13 to create a third rule with the following changes:
    - Name: **Public In**
    - Source: **Any**
    - Destination: **PhotoAppVM**
    - Services: **HTTP**
18. To the far right of each rule click the **GEAR**
19. Slide the **Slider** in the Dialog to **enable** logging
20. Click **APPLY**
21. Click **PUBLISH** to save and activate the rules

> ⚠ **Note:** Make sure to leave **All Uplinks** in the **Applied To** section.



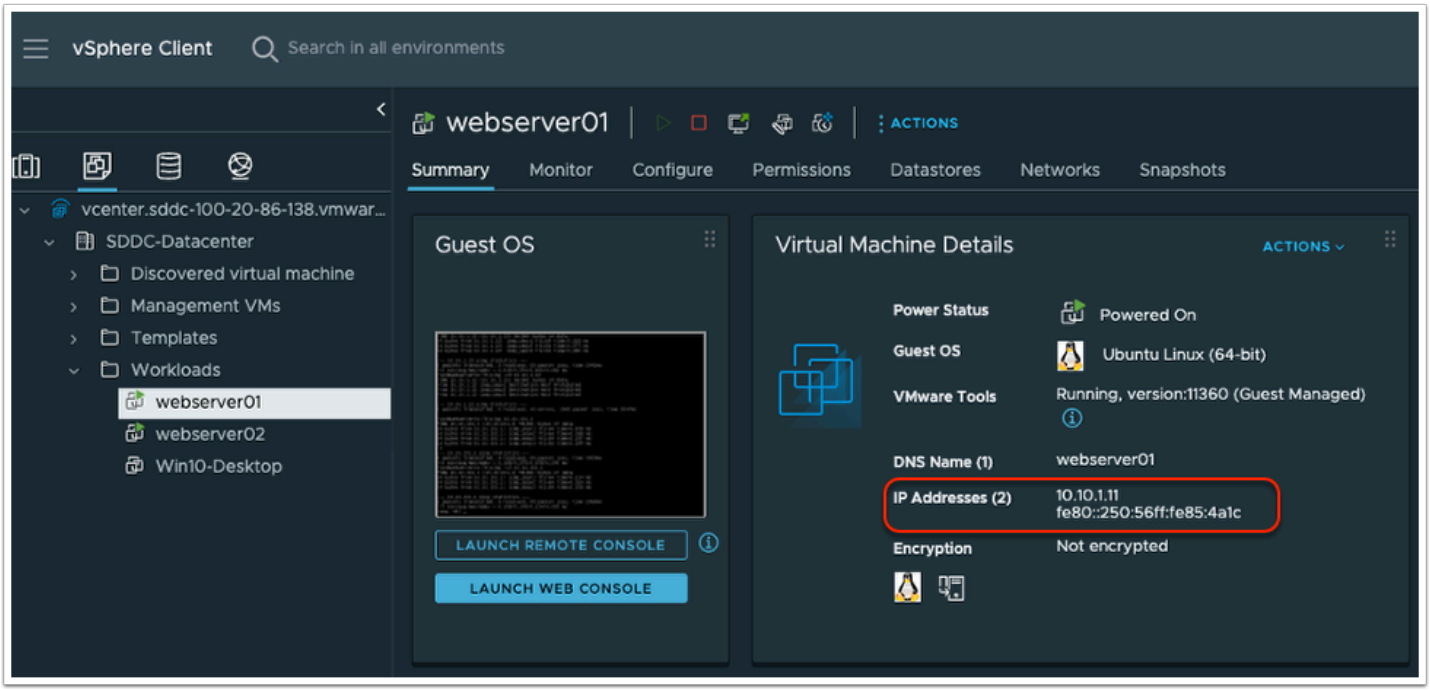
Task 3 - Request a public IP address

The PhotoAppVM (webserver01) currently has a private IP address (10.10.X.X) and thus not internet routable. to allow public internet access to the VM You'll first need to request a Public IP address. After the public IP address is provisioned, you will configure NAT to direct traffic from the public IP address to the private IP address of the PhotoAppVM.

You can request public IP addresses to assign to workload VMs to allow access to these VMs from the internet. VMware Cloud on AWS provisions the IP address from AWS.

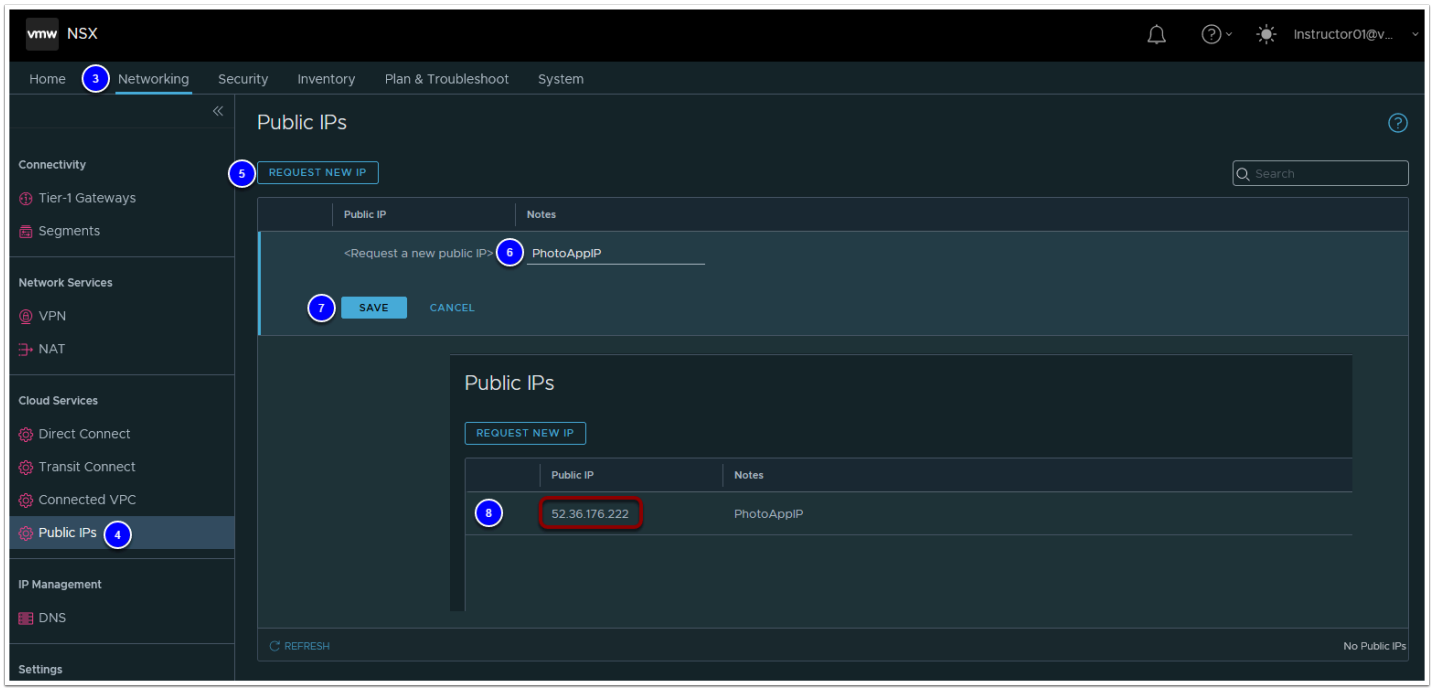
As a best practice, release the public IP addresses that are not in use.

1. In your vCenter interface for VMware Cloud on AWS, find your **Webserver01** VM you deployed, and ensure it has been assigned an IP address as shown in the graphic.
2. Take note of the IP address (Record the IP in the Excel Workbook provided)



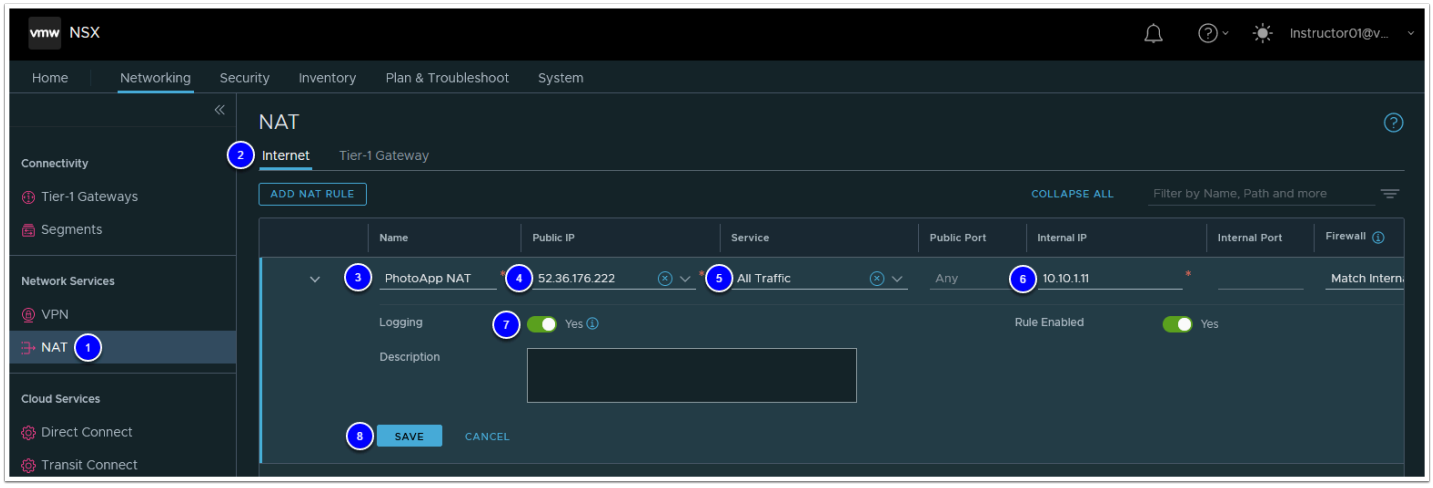
3. Go back to your NSX Manager UI on the **Networking** tab in order to request a Public IP address
4. Click **Public IPs** in the left pane
5. Click on **REQUEST NEW IP**
6. In the notes area type **PhotoAppIP**
7. Click **SAVE**
8. Take note of and record this **Public IP address**

The Public IP address will be used in the next task to setup Network Address Translation (NAT) for webserver01



Task 4 - Create a NAT Rule

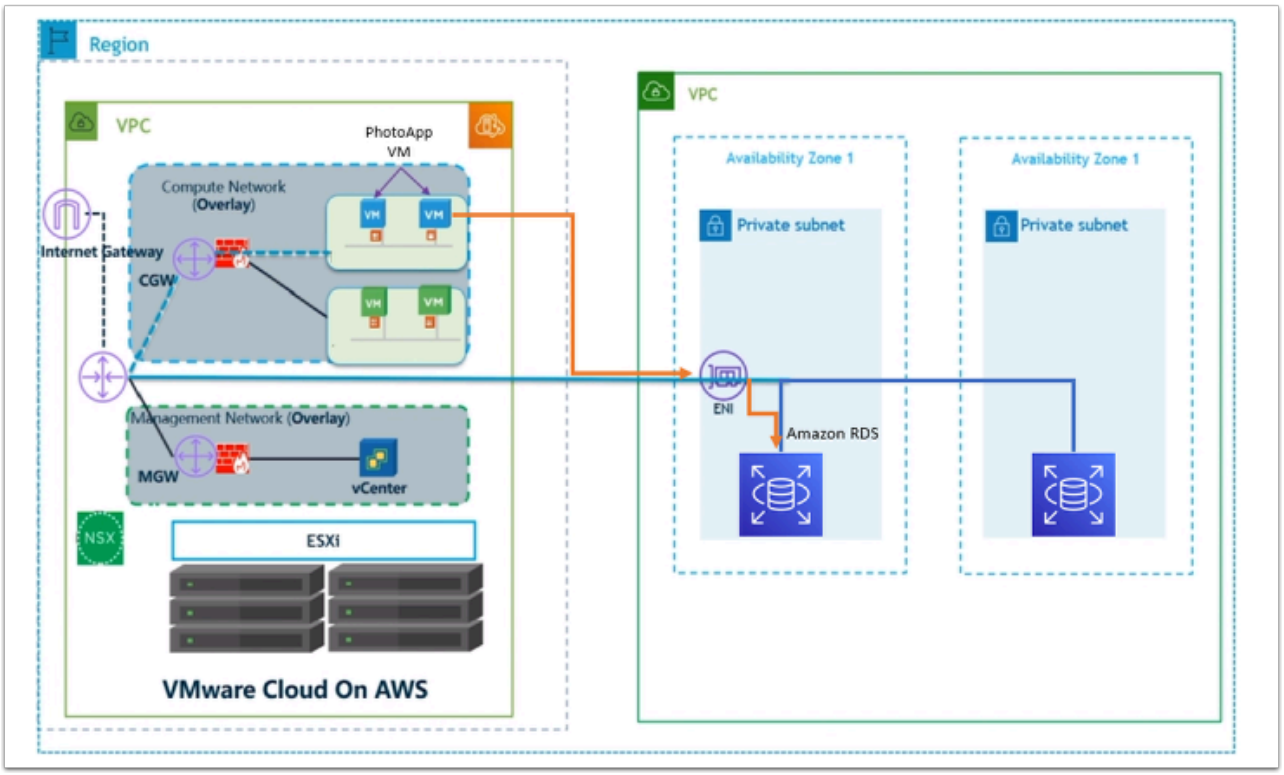
1. Click **NAT** in the left pane
2. Click **ADD NAT RULE**
3. Name: **PhotoApp NAT**
4. Public IP: From Task 3 (it should auto populate, but if it does not, select it)
5. Service: **All Traffic** (no change)
6. Internal IP: <IP address of your **Webserver01** VM you noted in task 2.1> See your Excel workbook for this IP if you've forgotten it
7. Logging: **YES** (Move the **Slider** to the right)
8. Click **SAVE**



Task 5 - View AWS RDS Instance and Security Settings

Amazon RDS makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security, and compatibility they need.

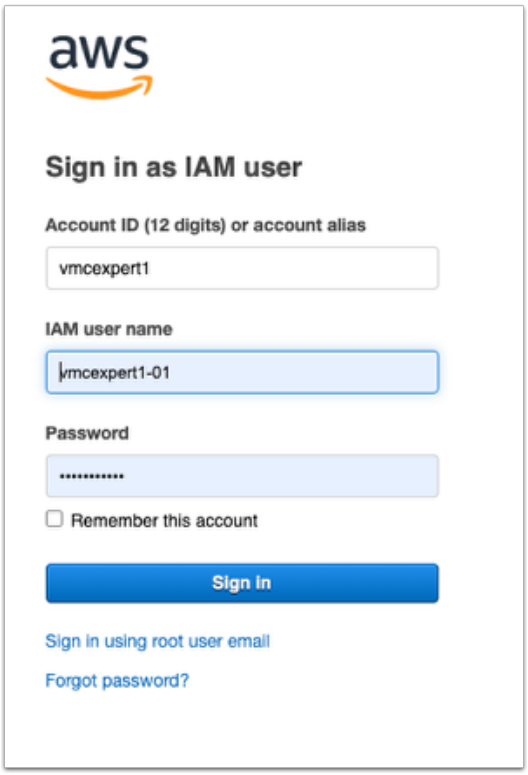
In this exercise, you will be able to integrate a VMware Cloud on AWS virtual machine to work in conjunction with a relational database running in Amazon Web Services (AWS) that has been previously set up on your behalf.



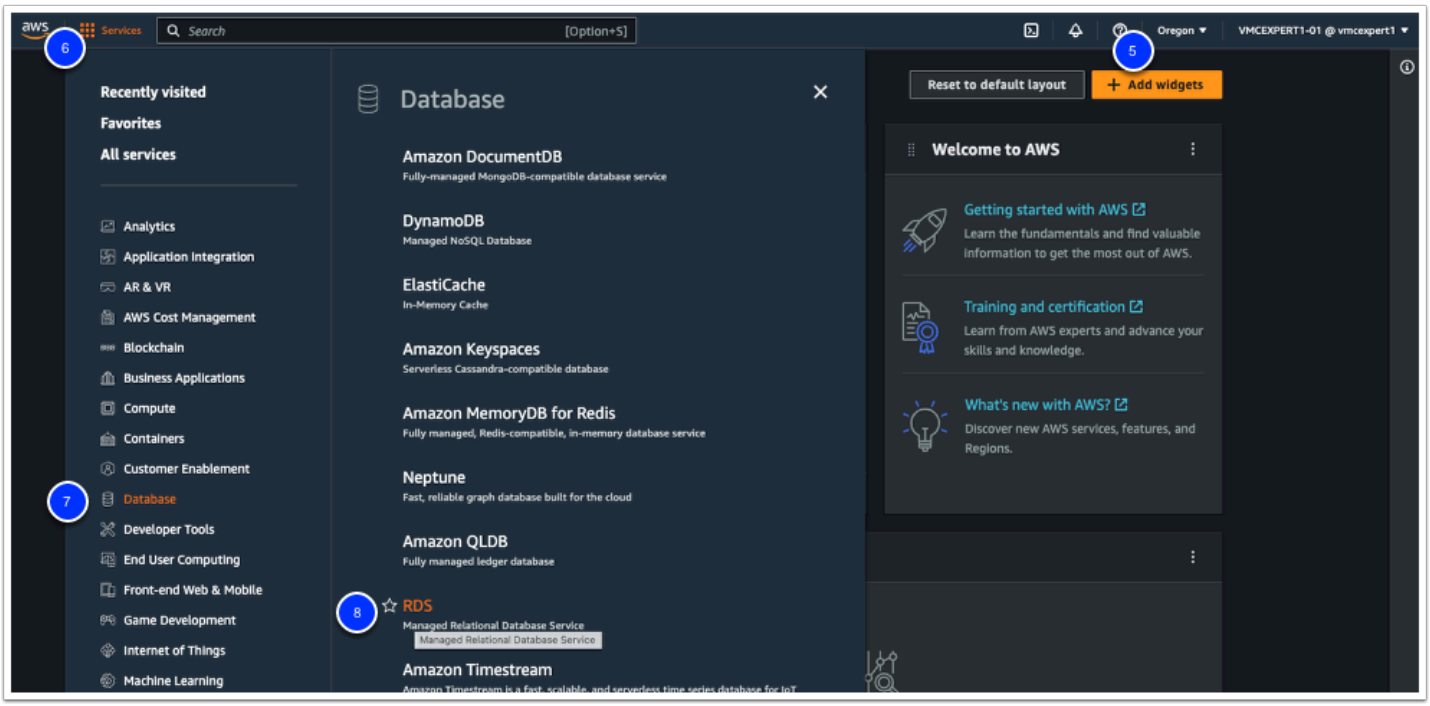
On your browser, open a new tab and go to: **https://vmcexpert{#}.signin.aws.amazon.com/console** where {**#**} indicates your AWS environment (1, 2 or 3)

The Credentials below are from the AWS Console portion of your student lab assignment sheet

1. Account ID or alias: **vmcexpert#** i.e vmcexpert1, vmcexpert2 or vmcexpert3
2. IAM user name: **VMCEXPERT#-XX**(where **#** is your Environment ID and **XX** is the number assigned to you)
3. Password: <**AWS Console PW provided By your instructor**>
4. Click **Sign In**

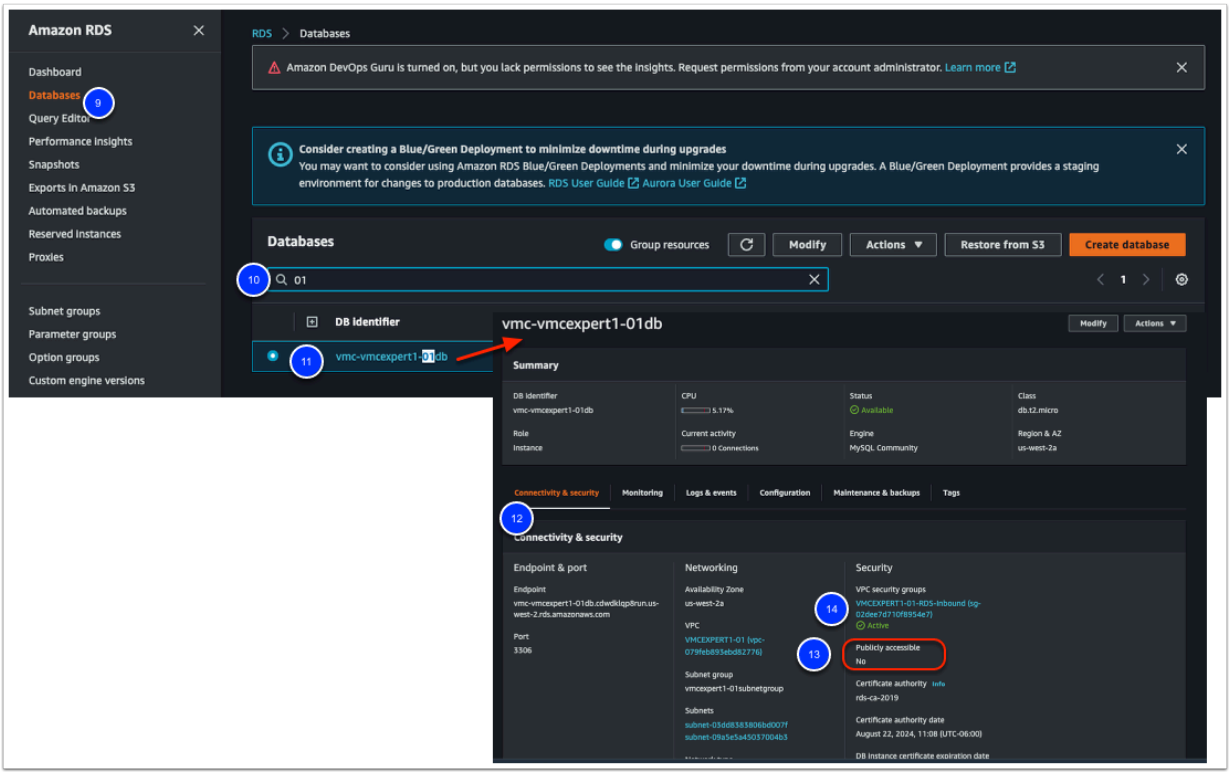


5. Make sure the region selected is **US West (Oregon) us-west-2** If you are using vmcexpert1 or vmcexpert2 environments or chose **Europe (Frankfurt) eu-central-1**, if your using vmcexpert3 or your instructor instructs you to do so.
6. Expand the **Services** drop down
7. Select **Database**
8. Select **RDS**

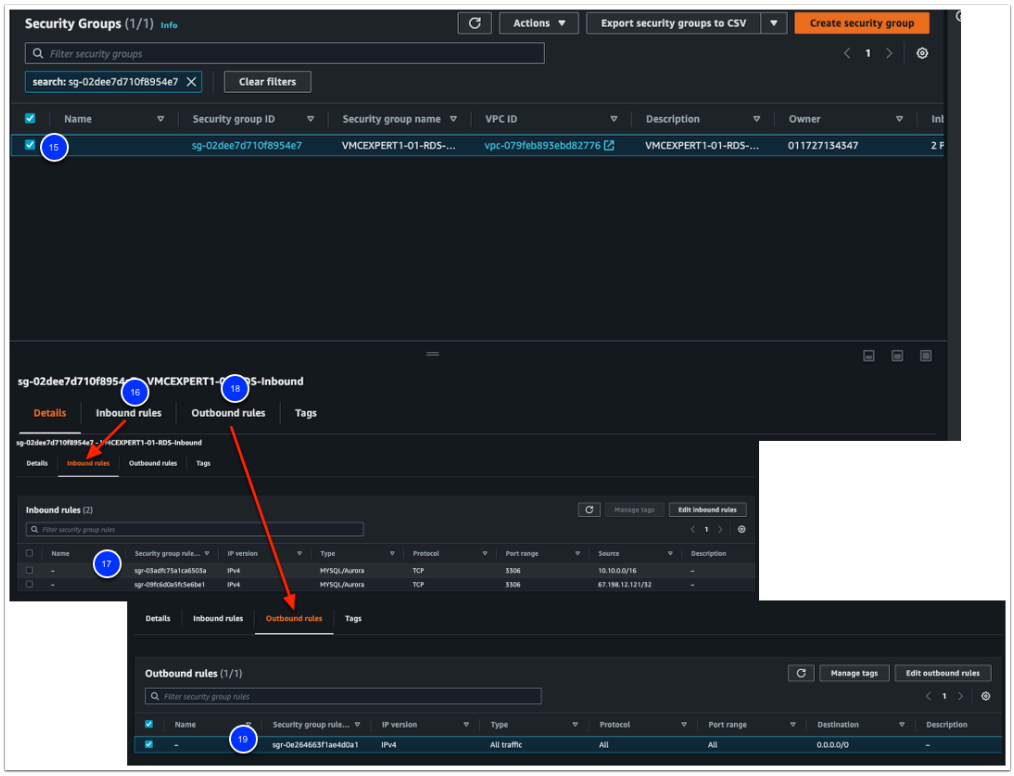


9. In the **Amazon RDS** left pane click on **Databases**
10. Search for your student number (**i.e. 01 through 31**)
11. Click the blue text under the **DB Identifier** column for the RDS instance that corresponds to your designated student number
12. Ensure that you are on the **Connectivity & Security** Tab.
13. Look inside the **Security subsection** at the **Public Accessibility** details**.** You may need to scroll down
14. Note the name and click on the link under **VPC Security Groups**

> Note the RDS instance is not publicly accessible, meaning this instance can only be accessed from within AWS.



15. Check the box next to your **security group** ie. VMCEXPERT#-XX-RDS-Inbound (may not match your student number).
16. Click on the **Inbound rules** tab in the pane below
17. Review the Inbound rules, and note that your PhotoAppVM should be allowed access to TCP Port 3306
18. Click **Outbound rules** tab
19. You can see **All traffic** (internal to AWS) is allowed; this includes your VMware Cloud on AWS SDDC logical networks.



Note: VMware Cloud on AWS establishes routing in the default VPC Security Group, RDS can leverage this or create its own

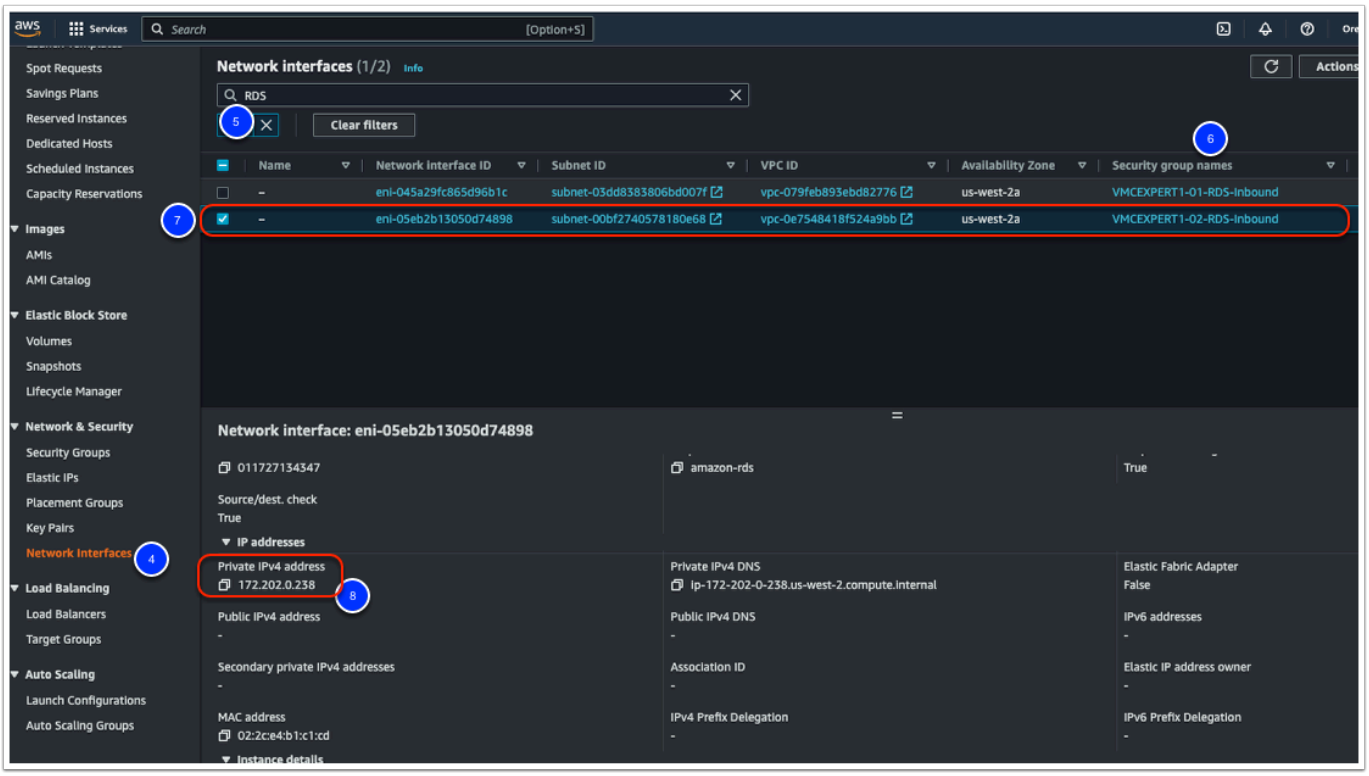
Task 6 - View the AWS RDS ENI Settings

AWS Relational Database Service (RDS), also creates its own Elastic Network Interface (ENI) for access which is separate from the ENI created by VMware Cloud on AWS.

1. Click on the **Services** drop down to go back to the Main Console
2. Click on **Compute**
3. Click on **EC2**

All Student environments belong to the same AWS account, therefore, hundreds of ENI's may exist. We will search for RDS to trim the results.

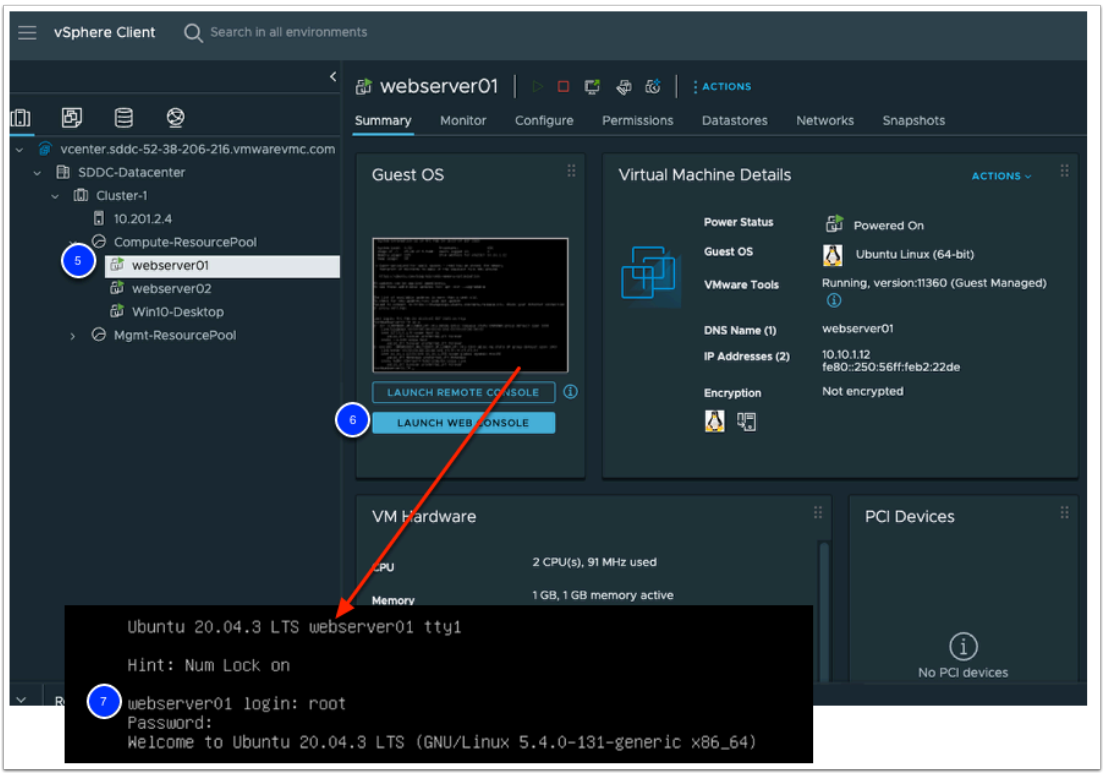
4. Under **Network & Security** in the left panel click on **Network Interfaces.** (you may have to scroll down)
5. Type **RDS** in the search area and press Enter to add a filter
6. Expand the **Security Group name** column to see the names
7. Find your **VMCEXPERT#-XX-RDS-Inbound** security group corresponding to your student number and check the box on the left (Don't click on the blue link)
8. Once selected, look in the details pane below to find the **Private IPv4 address.** (you may have to scroll down)
9. Copy this address to your notes for the next step



Task 7 - Configure & Test the PhotoApp against the AWS

You will now access the PhotoApp and update it's Database Connection (DSN - Data Source Name) by pointing it to the RDS instance. Once this is done you'll test the app by uploading some photos into the gallery.

1. Click the **VMC on AWS** browser Tab and Click the **Open vCenter** button
2. Click **Show Credentials**
3. Copy the **password** and click **Open vCenter**
4. **Log into vCenter as:**
   - Username: **Cloudadmin@vmc.local**
   - Password: **{Your_Saved_CloudAdmin_Password_From_Step_3}**
5. Click on **webserver01** in the Inventory on the left (you may have to expand the inventory)
6. Click on **Launch Web Console**
7. In the **webserver01** browser tab, log into the Virtual Machine as:
   - Login: **root**
   - Password: **VMware1!**



We will now modify the Lychee environmental file. This file will be used later to connect to the RDS database and configure the required database tables

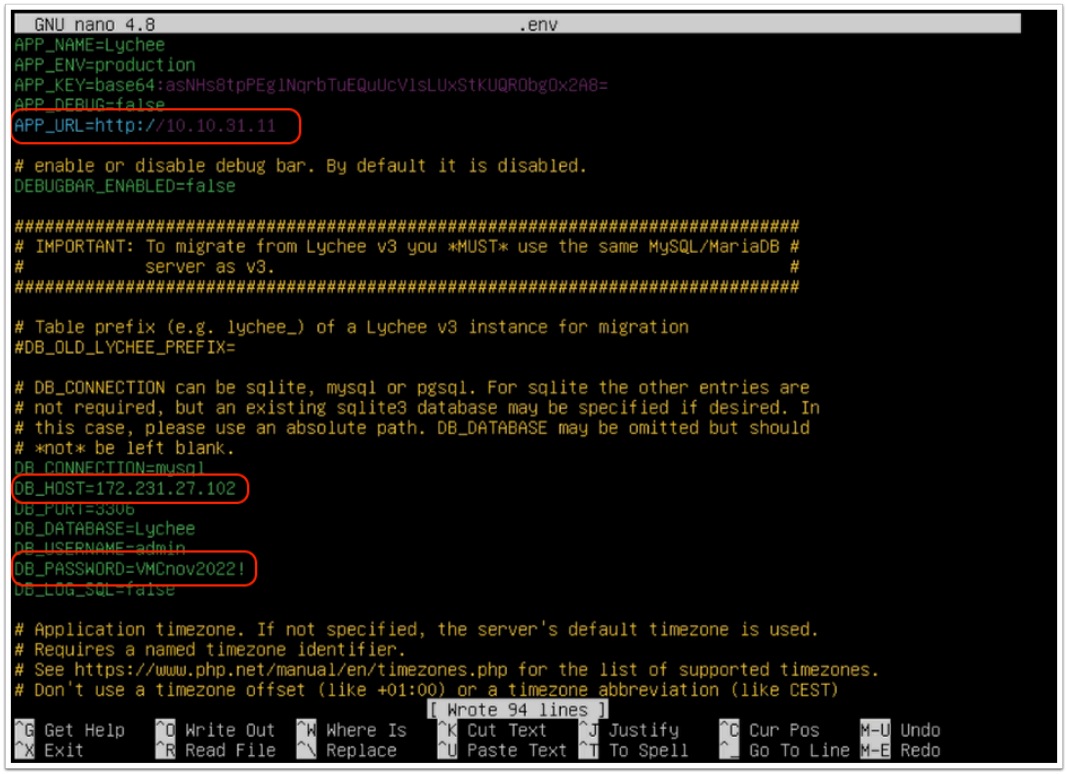
7. In the **webserver01** Console, navigate to the Lychee application directory make a copy of the .env file and open it in an editor by typing the following commands

```
cd /var/www/html/Lychee
cp .env .env_orig
nano .env
```

Click to copy

9. With the environment file opened, modify the following values:
   - **App_URL: {Replace_the_IP_With_the IP_address_of your_WebServer01}**
   - **DB_HOST: {Replace_the_IP_with_the_IP_address_of_the_RDS_DB} NOTE: This IP was recorded in Task 6, Step 9**
   - **DB_PASSWORD: {AWS_Console_Password_Provided_By_your_Instructor}**

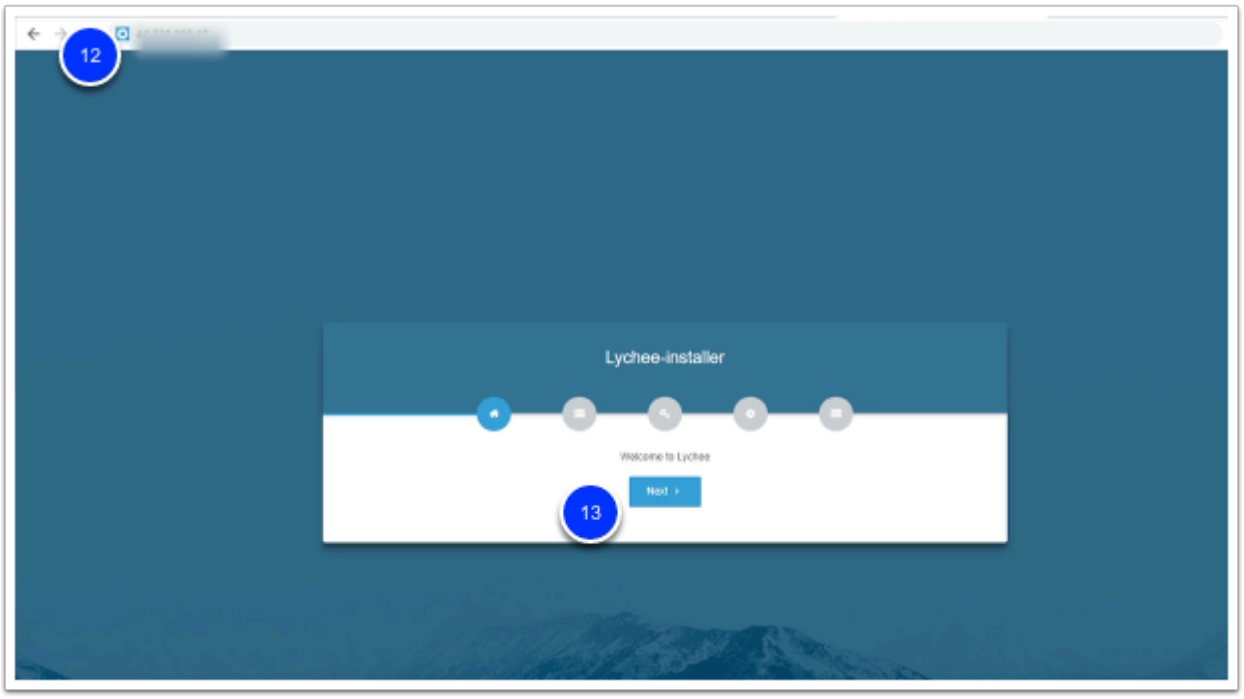
The DB Password is the same one you used to log in to the AWS Console in Task 5



10. Press **CTRL+O**, then Enter to save the changes
11. Press **CTRL+X** to close the file

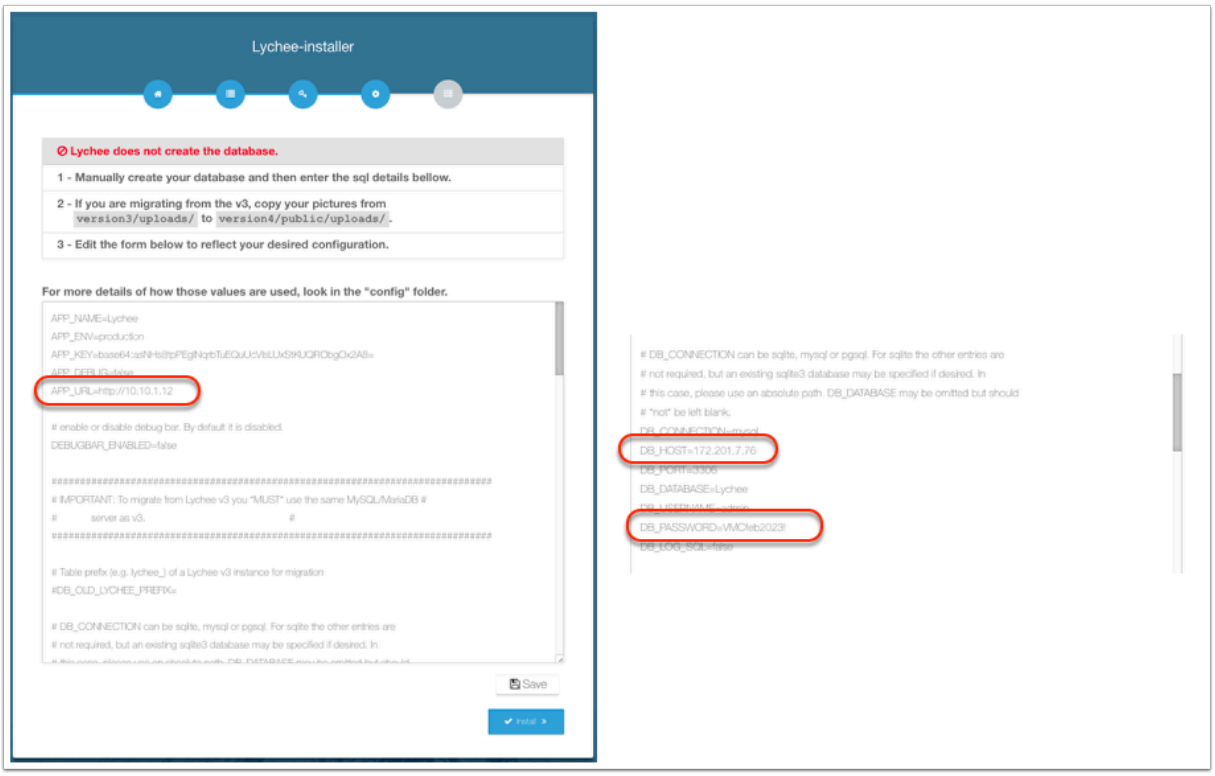
We will now run the configuration wizard, to confirm the configuration values, link the application to the RDS Database and create the necessary database tables

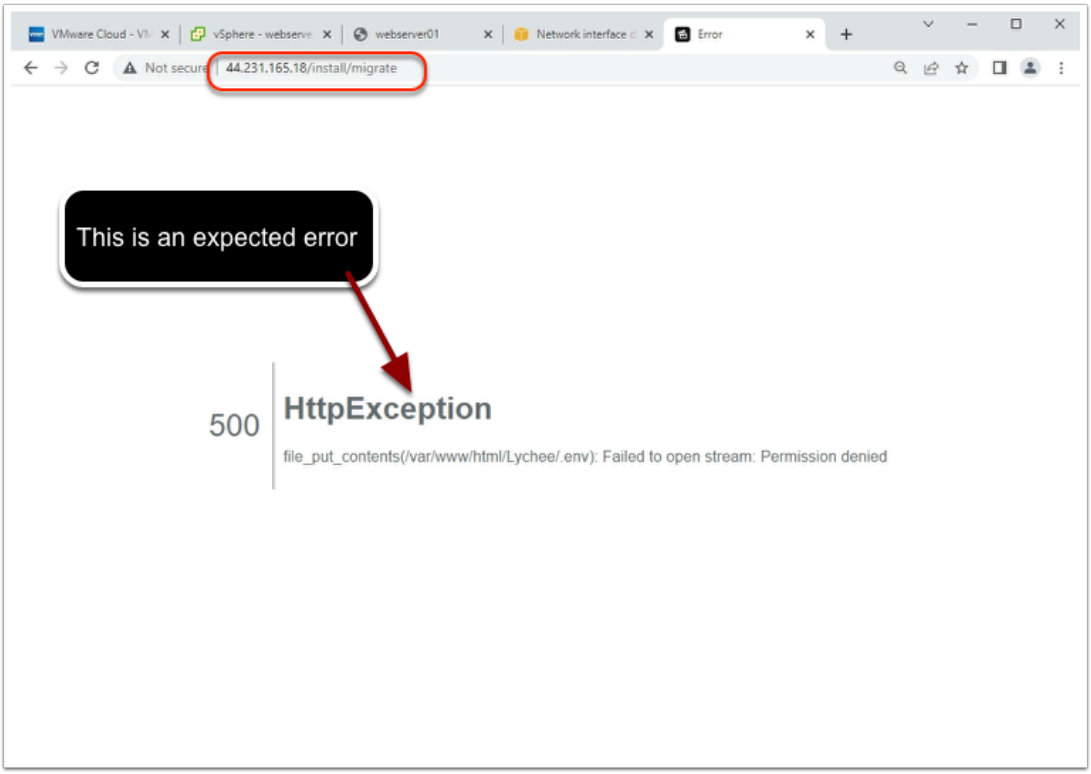
12. In a new browser tab, type/paste in the Public IP Address you requested in Task 3 Step 6 and used for your NAT rule in Task 4
13. In the Lychee Installer Wizard, Click **Next**



14. Click **Next** on the requirements page
15. Click **Next** on the Permissions page
16. Review the values in the Environment page to confirm the values you entered in **step 9** are correct (you may have to scroll to see all the values)
17. Click **Install** (An error message is expected, as we are not migrating an old database.)

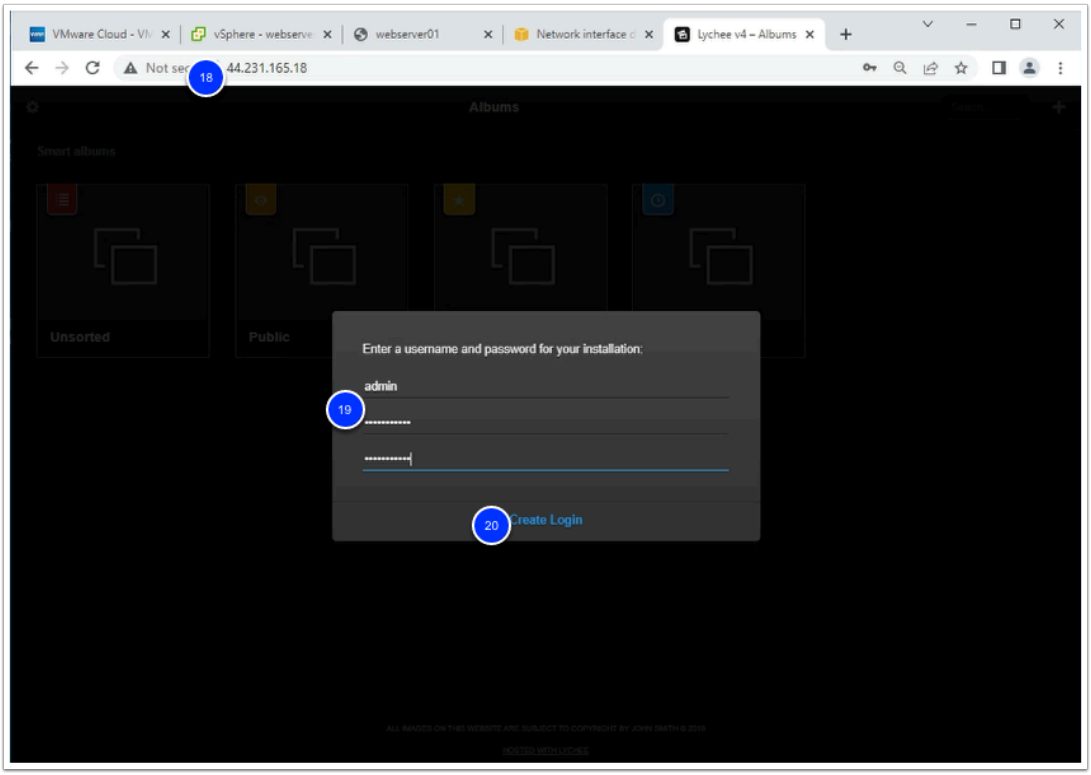
> ⚠ If any of the values are incorrect return to Steps 8 through 11 to correct the erroneous values and repeat steps 12 onward





Now that we have Lychee configured, the first time we connect it will ask us to create an account.

18. From your smartphone, tablet or In a new browser window type/paste in the Public IP Address of your webserver01. This is the IP requested in Task 3 Step 8
19. Create an Application admin account by entering the following:
    - **New Username: admin**
    - **New Password: {AWS_Console_Password_Provided_By_your_Instructor}**
    - **Confirm Password:{AWS_Console_Password_Provided_By_your_Instructor}**
20. Click **Create Login**



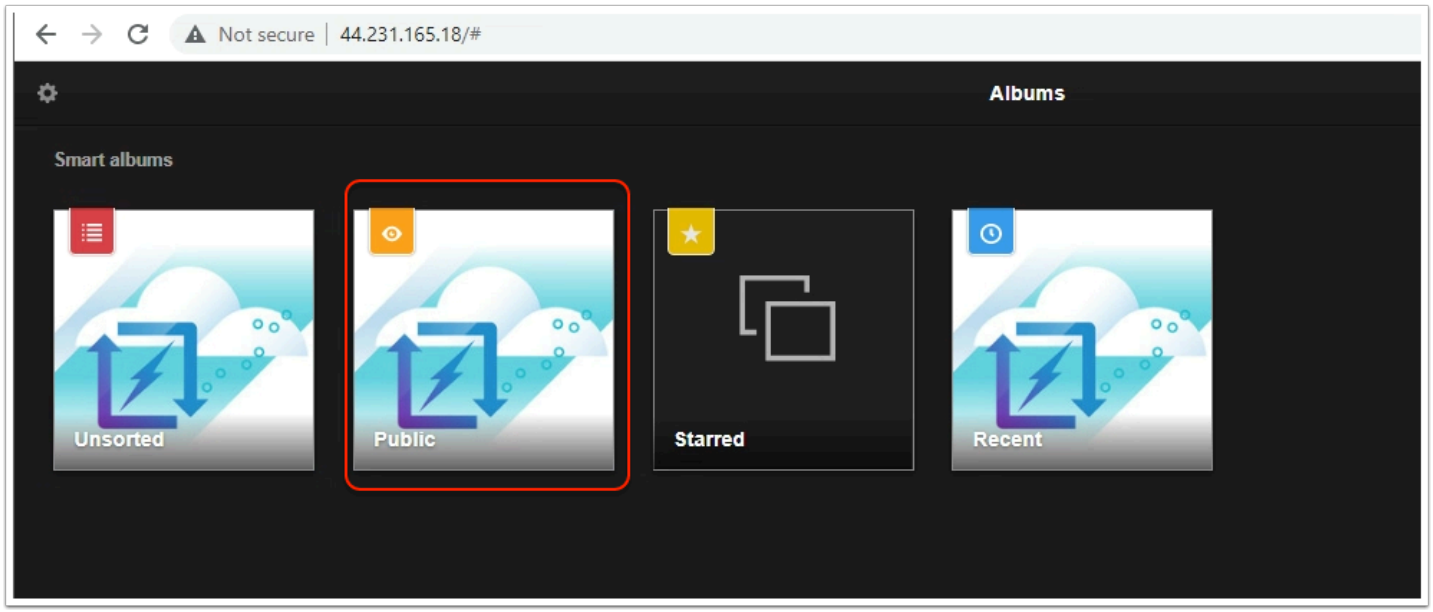
21. In the **webserver01** VM Console Browser tab, Type the following command

```
cd /var/www/html/Lychee/public/uploads/thumb
ls
```

Click to copy

Next we will upload some images and you'll notice changes to the file system. As images are stored a directory and thumbnail for each image is created.

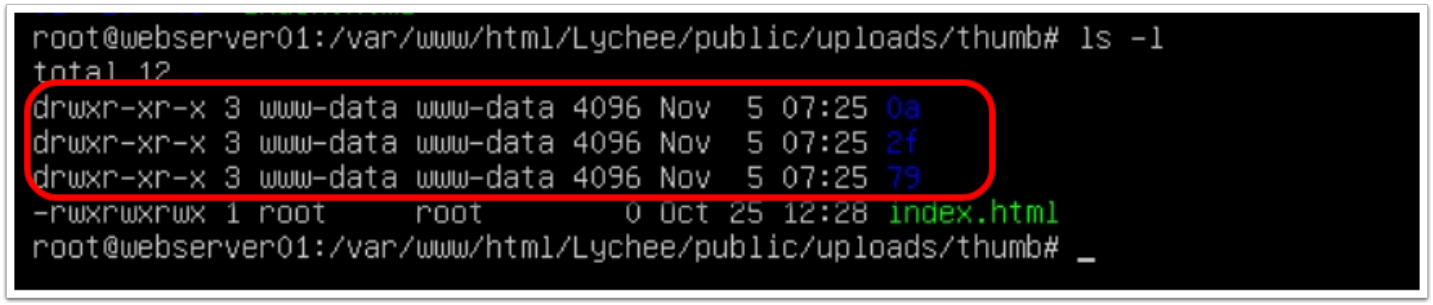
22. In the Lychee application browser window, click the **Public** folder
23. In the upper right-hand corner, Click the "**+**" icon
24. Click **Upload Photo**, and upload a few images



25. In the webserver01 VM Console Browser tab, Type the following command to confirm the files were stored on the local filesystem

```
ls -l
```

Click to copy



Congratulations, you have successfully logged in to the photo app, configured it to use the AWS RDS Database running in the Connected VPC and uploaded some images.

NOTE: The RDS MySQL DB is not used to store the photos. All photos are stored on the VMs local file system. The RDS stores all metadata about uploaded photos. Such as:

- Folder location
- whether or not the image was tagged as a favorite
- Public vs Private Photo
- etc..

This configuration is great but would prove problematic when the need arises to scale the application. In addition to using a centralize DB for metadata, we may want to store the images in a central repository as opposed to local storage. In the Additional (Optional) tasks you'll see how you can use an Amazon EFS and ALB to scale the application.

Conclusion

In summary, the front end (web server) is running in VMware Cloud on AWS as a VM, the back end which is a MySQL database is running in AWS Relational Database Service (RDS) and communicating through the Elastic Network Interface (ENI) that gets established upon the creation of the SDDC.

You have completed the required AWS Integration Lab.

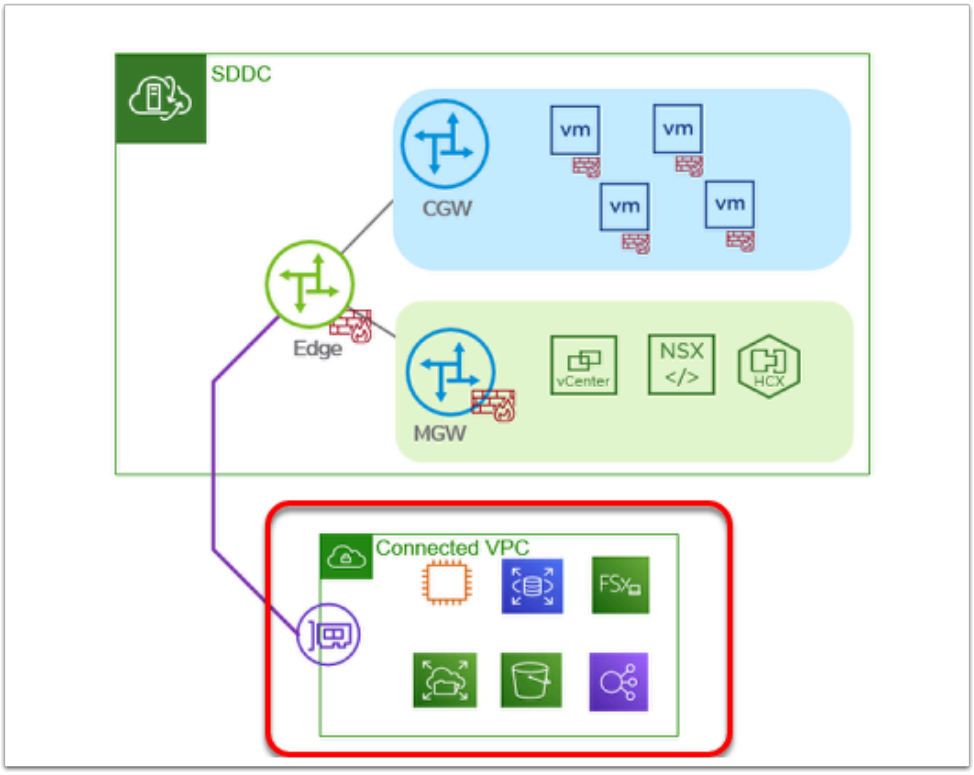
ADDITIONAL LABS

VMware Cloud on AWS enables you to have a hybrid cloud platform by running your VMware workloads in the cloud while having seamless connectivity to your AWS native services.

The integration which VMware and AWS have created allows for these services to communicate, for free, across a private network address space for services such as EC2 instances, which connect into subnets within a native AWS VPC, or with platform services which have the ability to connect to a VPC Endpoint, such as S3 Storage.

In these optional lab exercises we will build on what we learned from the previous lab tasks by configuring integration with other Native AWS Services such as:

- Amazon Elastic File System (EFS)
- Elastic Load Balancing (ELB)



When you deploy an SDDC on VMware Cloud on AWS, it is created within an AWS account and VPC dedicated to your organization and managed by VMware. You must also connect the SDDC to an AWS account belonging to you, referred to as the customer AWS account. This connection allows your VMC SDDC to access AWS services belonging to your AWS VPC account.

Additional Lab 1 - Consuming EFS Storage in VMC on AWS

Although the VMware Cloud on AWS SDDC Provides a multi-TB datastore for storing Virtual Machines and supporting files, there may be specific criteria of application data that you want running on your NVMe drives, and other data that is classified as 'lower tier'. If that is the case, one of the options you have with VMware Cloud on AWS is to leverage Amazon

Elastic File System (EFS) for additional data. You can think of EFS as a very simple and easy to use Network File Share. A single EFS can be added to multiple VMs if you choose to do so, or to single VM.

Amazon supports this for Linux operating systems only at this time.

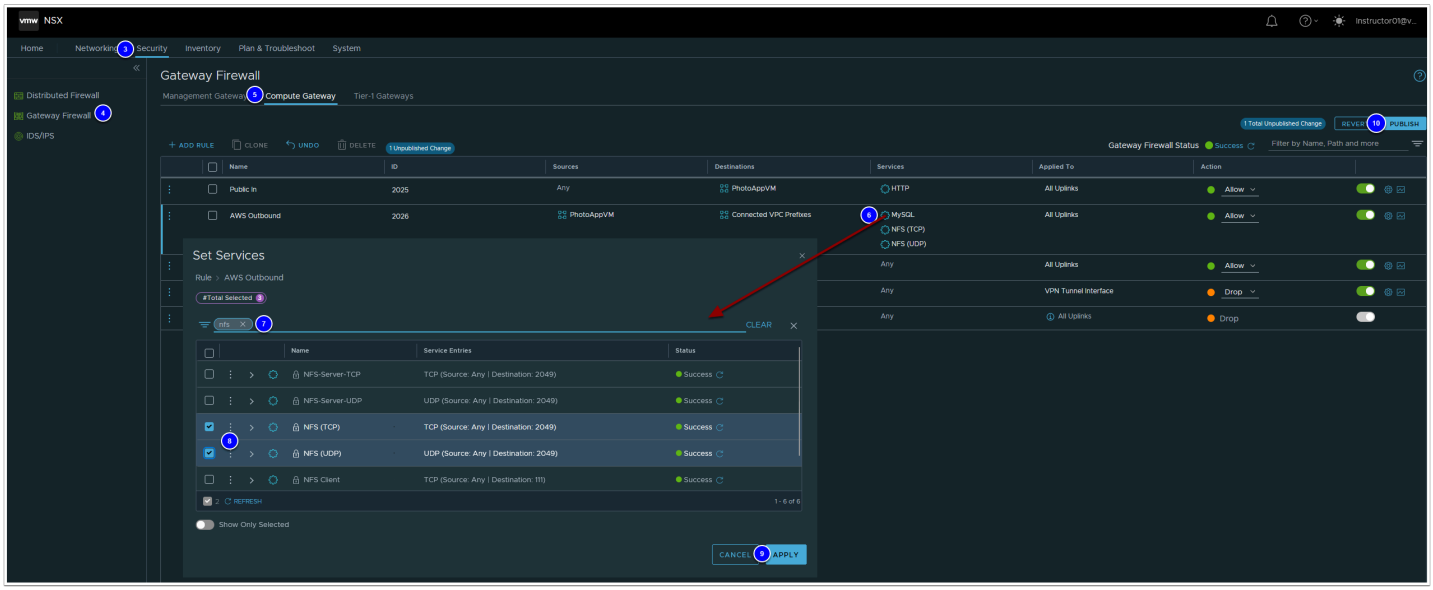
⚠ Prerequisites:

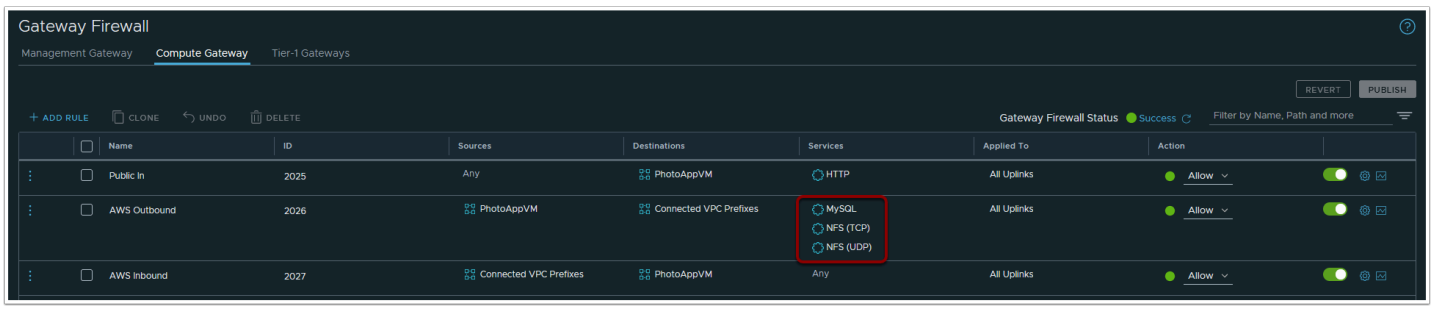
- Lab 2, All Tasks
- Lab 3, All Tasks

Task 1 - Configure VMC on AWS Gateway Firewall Rules

Because all traffic over the ENI is denied by default, you need modify the gateway firewall to allow the required traffic to flow uninterrupted. For this reason we will modify the "**AWS Outbound**" rule on the Compute Gateway to allow access to EFS over the ENI.

1. If your NSX Manager UI tab is active then go to step #3. If you already closed NSX Manager tab then Select your SDDC, if you aren't currently within it, then click **View Details**
2. Click the **OPEN NSX MANAGER** button and click **ACCESS VIA THE INTERNET** to connect to NSX Manager UI. Wait till page with NSX Manager will be loaded and you will see **Home - Overview** dashboard.
3. In the NSX Manager UI interface click the **Security** tab
4. Click **Gateway Firewall**
5. Click **Compute Gateway**
6. Hover over the **Services** field of the **"AWS Outbound"** Rule and Click the **Edit** (Pencil Icon)
7. In the Search field of the Set Services Dialog Type **NFS** & Press Enter
8. Select **NFS(TCP)** & **NFS(UDP)**
9. Click **Apply**
10. Click **Publish**

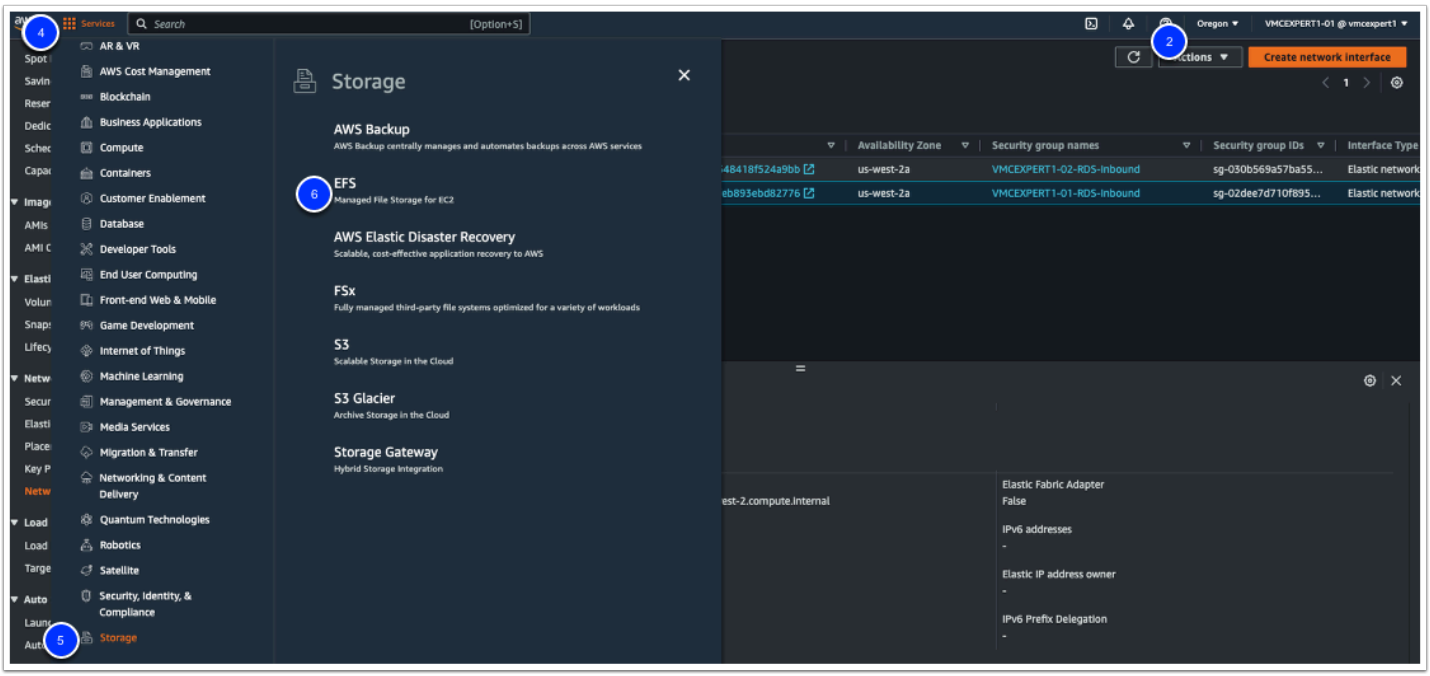




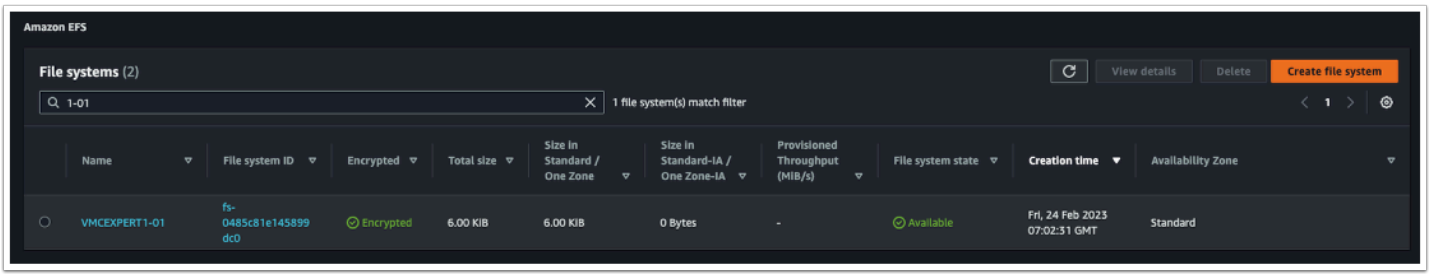
Task 2 - Review the EFS Settings in AWS

We will now access the AWS Console to confirm the existence of a pre-deployed EFS. We'll also need to identify the IP address of the EFS, as we'll need to to create the mount in your Virtual Machine.

1. Log into the AWS console using the AWS console link and credentials in the student lab assignments worksheet.
2. Confirm you are administering services in the **Oregon** Region (top right corner drop down)
3. If not, Click the drop-down and select **US West (Oregon) us-west-2** (If you are using vmcexpert1 or vmcexpert2 environment)
   select **Europe (Frankfurt) eu-central-1** (if you are using vmcexpert3)
4. Click the **Services** drop down
5. Select **Storage**
6. Select **EFS**

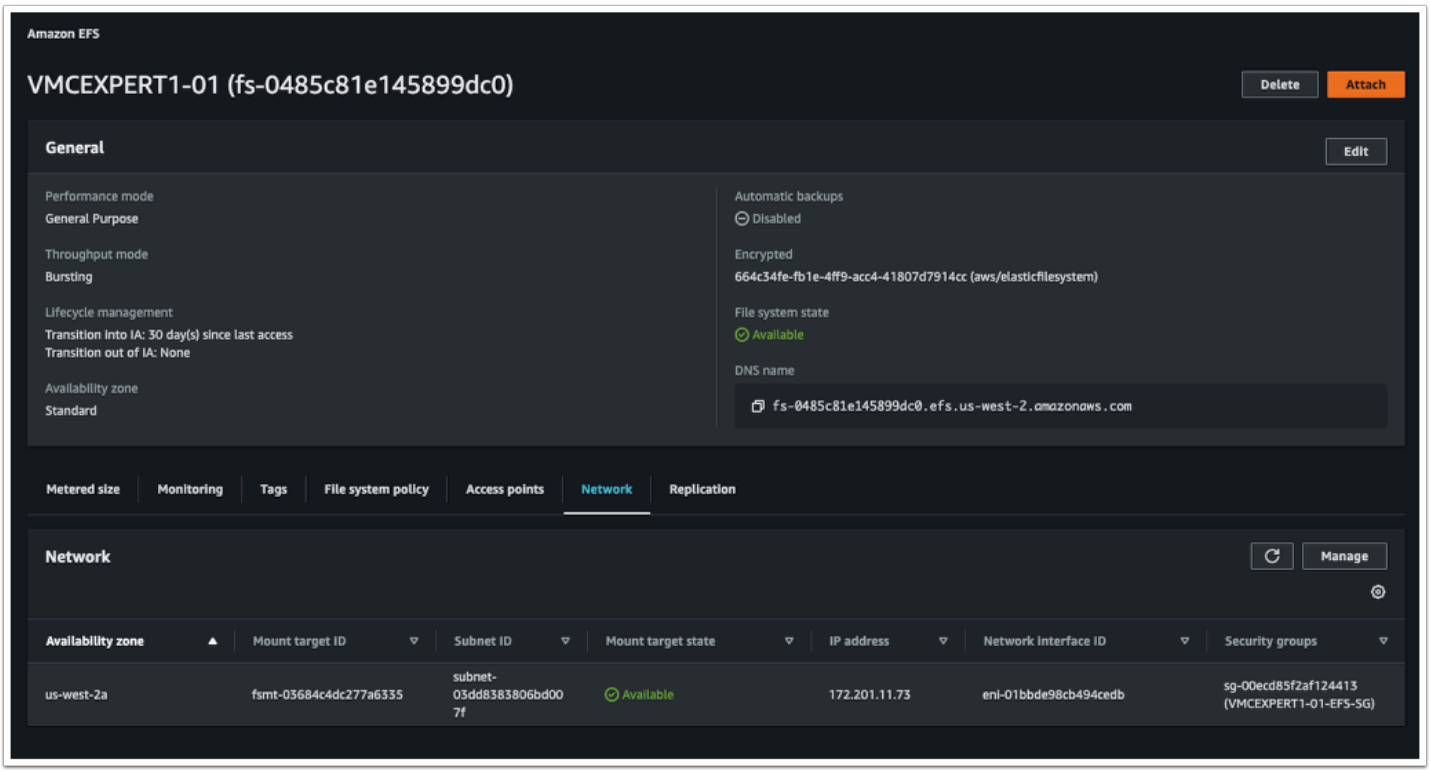


7. In the list of file systems find your EFS **(VMCExpert#-xx**, where **#** is the Environment ID, and xx is your student number)
8. Click <**your EFS Instance> vmcexpert#-xx** text to view its details



9. Click the **Network** tab
10. Record the IP address of your EFS (e.g. 172.120.11.73)

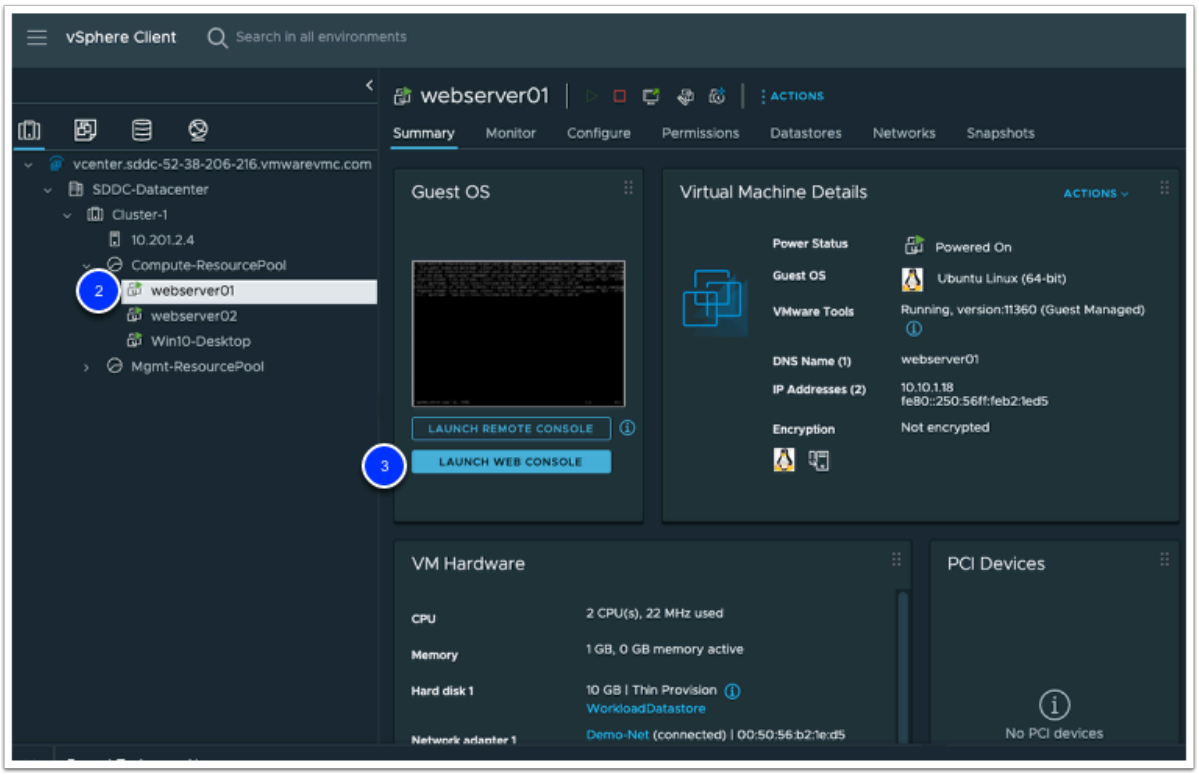
> You will need this IP to mount the share in your Webserver01 VM



Task 3 - Mount an EFS share in a VM running in VMC on AWS

1. If the browser tab to the SDDC vCenter is still open navigate to it. If not Open a new Tab and log onto the VMC SDDC vCenter.
2. Select **webserver01**
3. Click **LAUNCH WEB CONSOLE**
4. In the browser tab for **webserver01**
5. If needed, Log in as
    1. login: **root**
    2. password: **VMware1!**

NOTE: You can access the vCenter Information and login details from the settings tab of the VMC on AWS Console



6. At the shell prompt enter the following commands

> ⚠ Note, your current directory **must** be /var/www/html/Lychee/public for the prep-webserver-1.sh script to work correctly - make sure to run the cd command as shown:

```
cd /var/www/html/Lychee/public
./prep-webserver-1.sh {your_efs_ip}
```

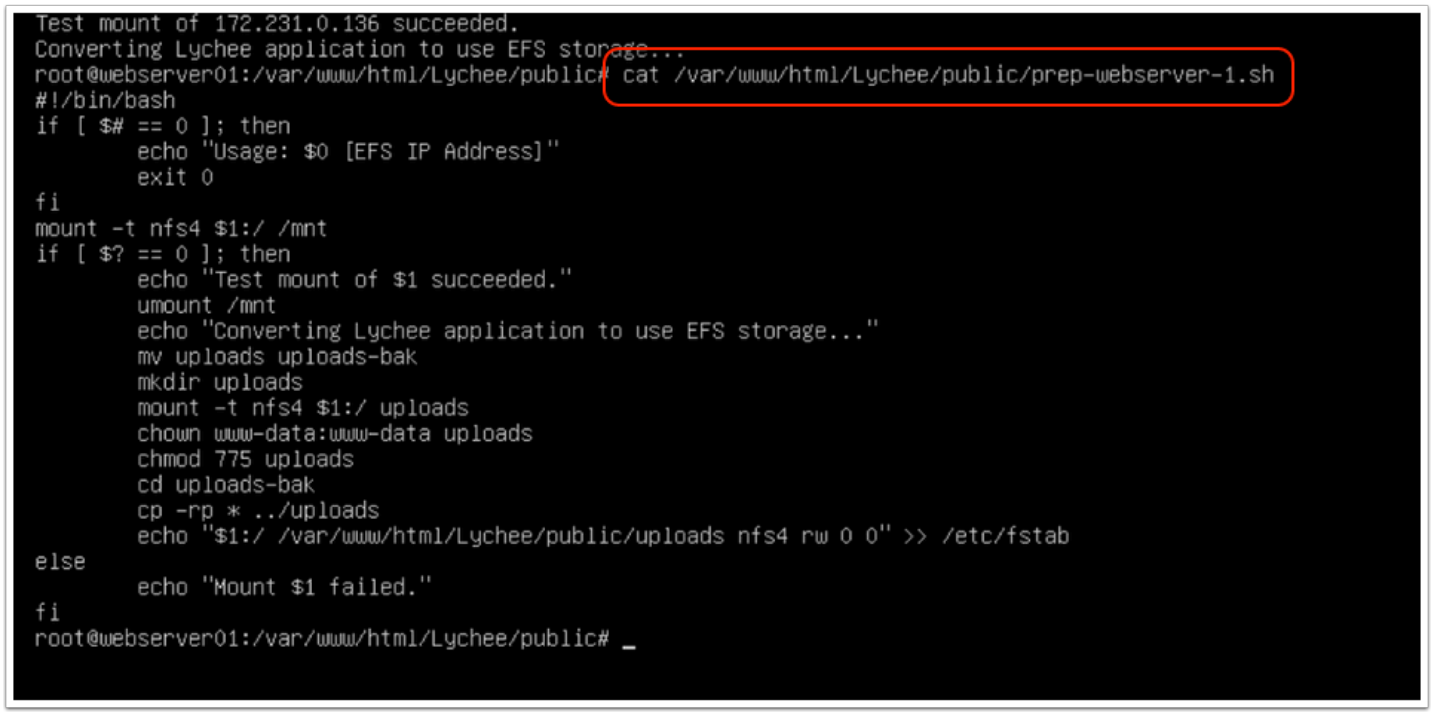
Click to copy

> This script converts the storage of the photo app from the local file system to an NFS share on AWS EFS

7. to view the operations performed by the script let's take a look at the script. Type the following command

```
cat /var/www/html/Lychee/public/prep-webserver-1.sh
```

Click to copy

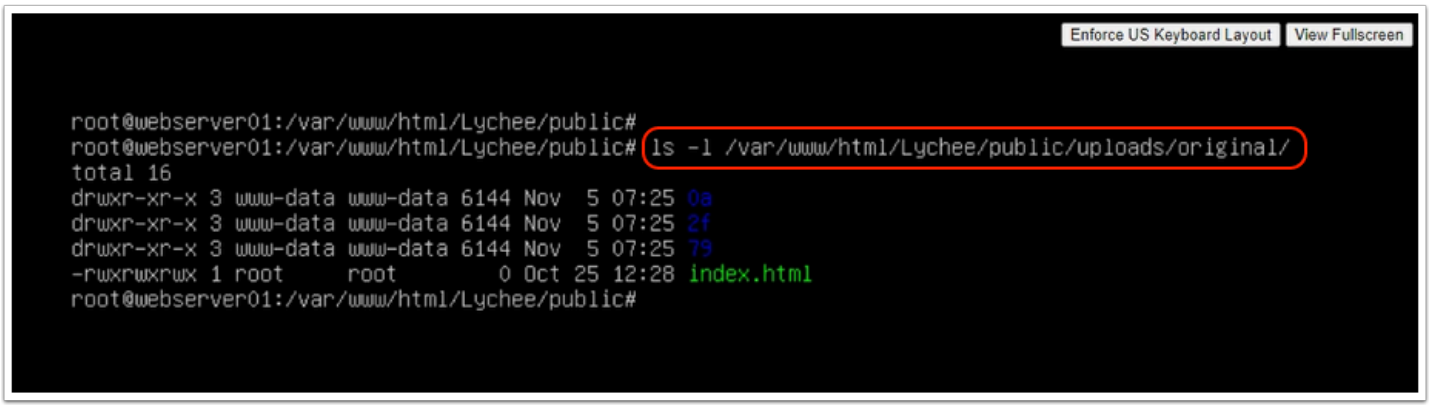


Now let's take a look at the NFS mount to confirm the images were copied

8. Type the following command:

```
ls -l /var/www/html/Lychee/public/uploads/original
```

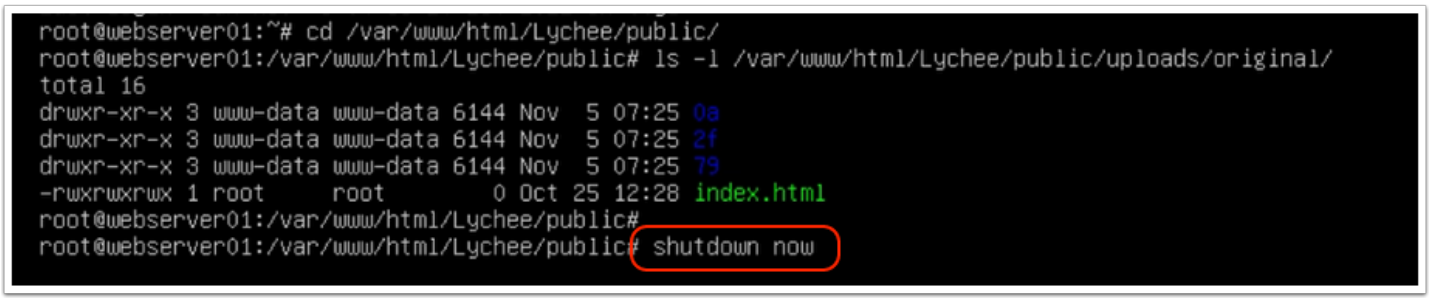
Click to copy



9. Shutdown the VM using the following command:

```
shutdown now
```

Click to copy



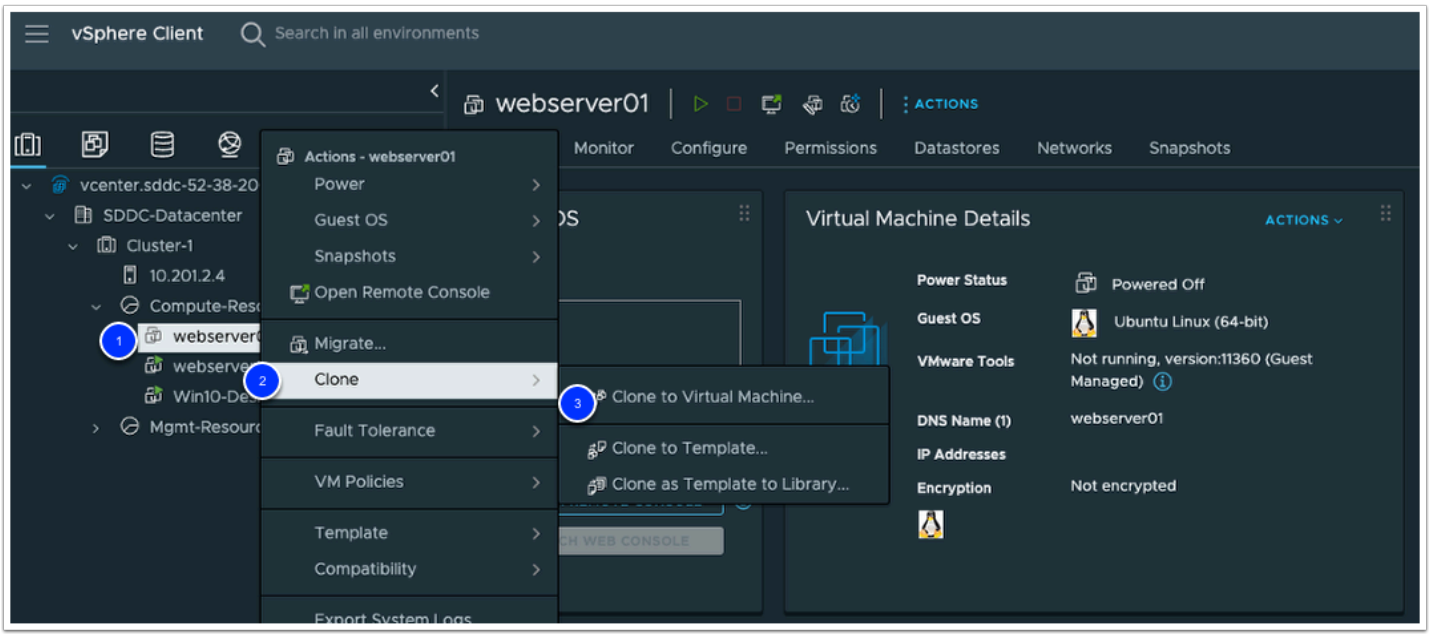
10. Close the **webserver01** browser tab

Task 4 - Clone Webserver01

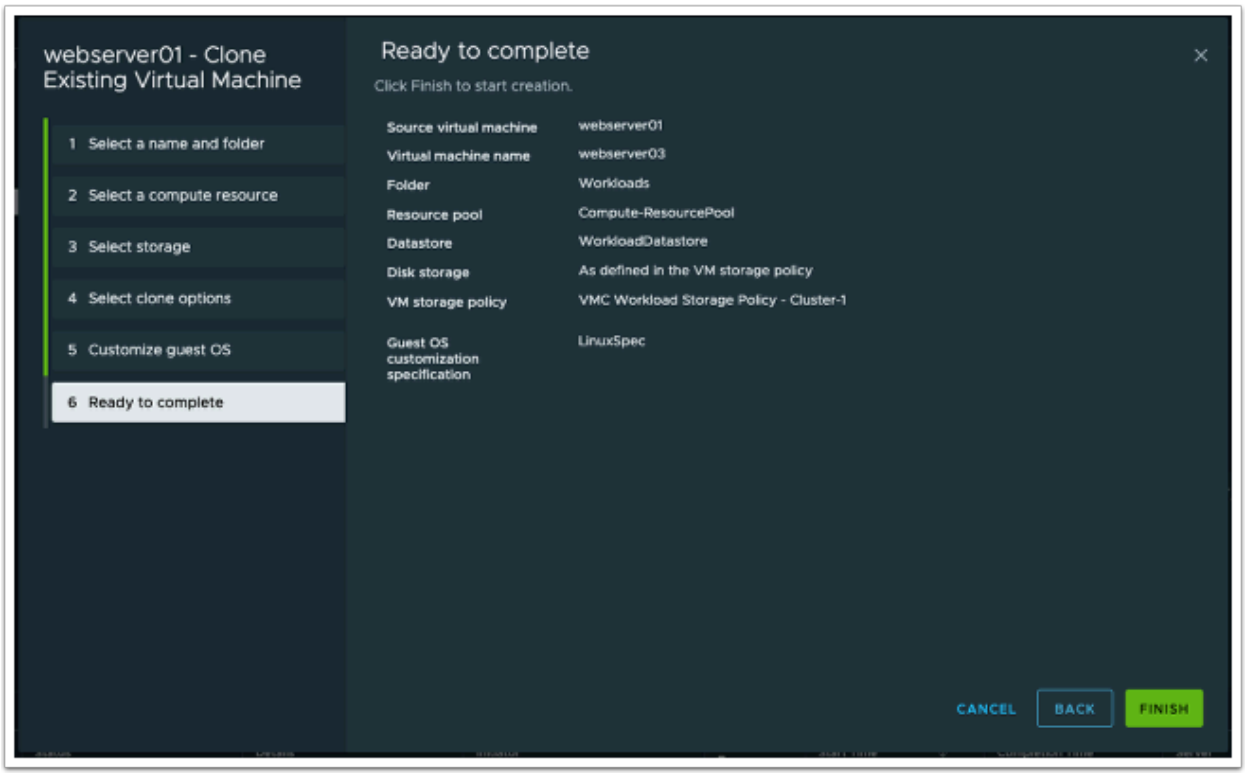
We will now clone webserver01 to create a new Virtual Machine "webserver03". We perform this task to confirm webserver03 continues to have access to the files in the same central repository as webserver01.

1. In the vSphere Client browser tab, Select and right-click **webserver01**
2. Select **Clone**
3. Select **Clone to Virtual Machine**

It should take a couple of minutes for the virtual machine to clone.



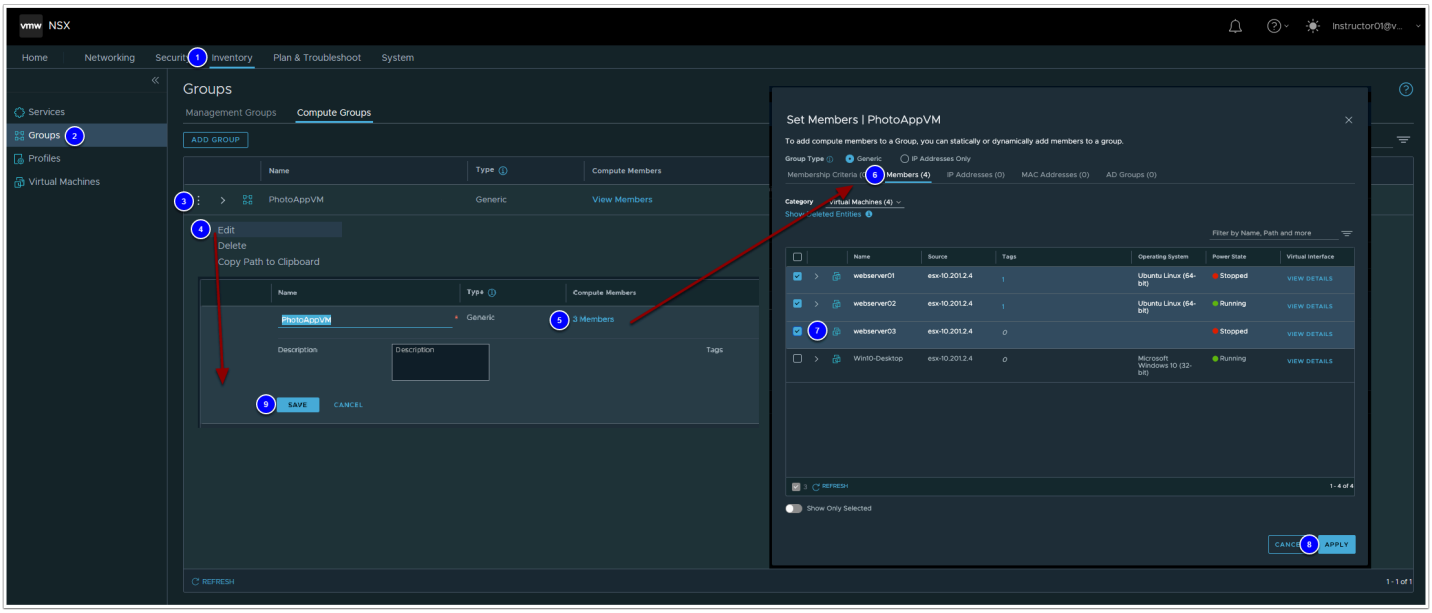
4. On the **Select a Name and Folder** page name the **virtual machine name** enter **webserver03**
5. Expand the **vCenter > SDDC-Datacenter >** and highlight **Workloads** folder
6. Click **Next**
7. On the **Select a Compute Resource** page select the **Compute-ResourcePool**
8. Click **Next**
9. On the **Select Storage** page select the **WorkloadDatastore**
10. Click **Next**
11. On the **Select Clone Options** page click the following check-boxes
    - **Customize the operating system**
    - Do not Select **Power on virtual machine after creation**
12. Click **Next**
13. On the **Customize Guest OS** page select the **LinuxSpec** customization specification.
14. Click **Next** to continue.
15. Review the information for accuracy and click **Finish** to clone the virtual machine.



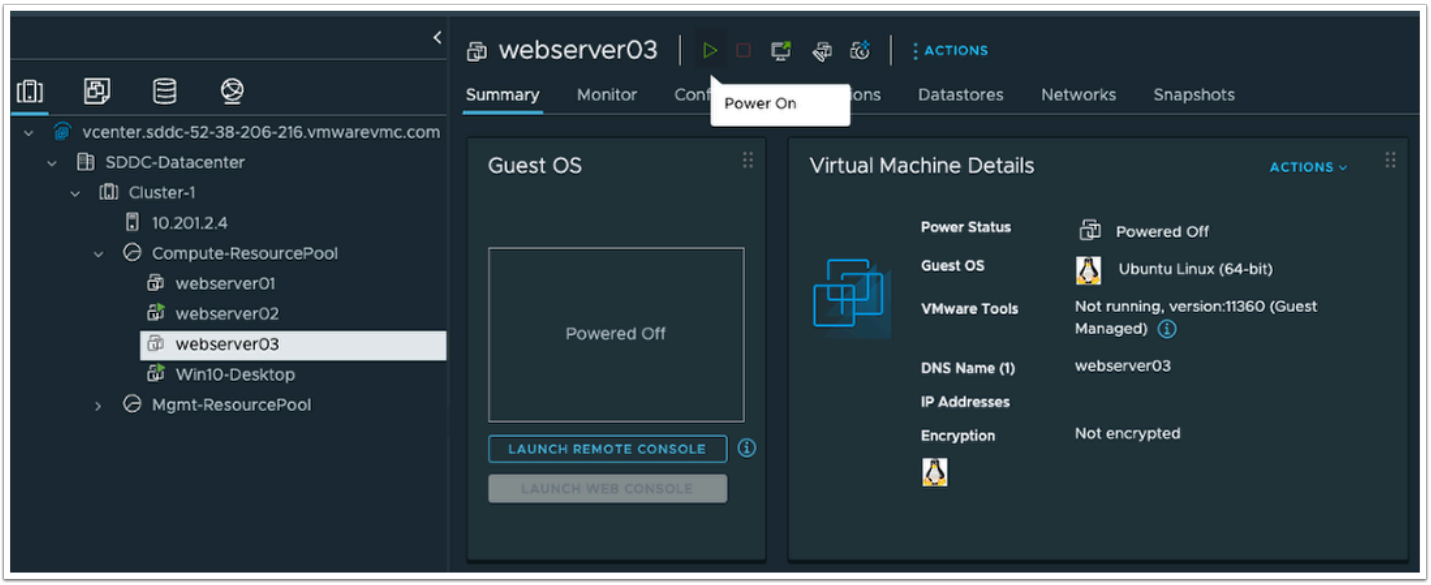
Task 4.1 - Add webserver03 to PhotoAppVM Group

In a previous task we created the PhotoAppVM Group which we used in the Gateway firewall rule. We need to add **webserver03** to this group. Doing so will add it to the firewall rule along with **webserver01**

1. In the NSX Manager UI browser tab, click the **Inventory** tab
2. Click **Groups**
3. Click the **3 vertical** dots next to the **PhotoAppVM** Group
4. Click **Edit**
5. Click **Members**
6. Click the **Members** tab
7. Select **webserver03** to add it to the group
8. Click **APPLY**
9. Click **SAVE**



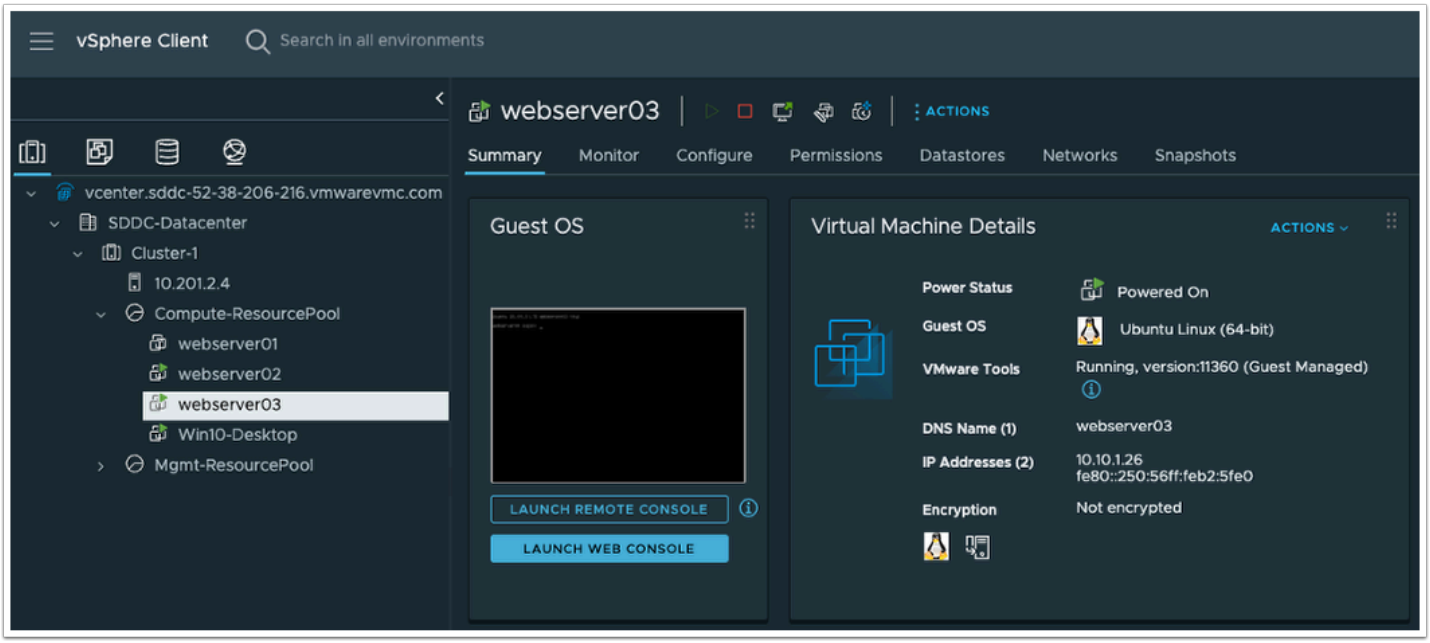
10. The clone of **webserver03** should be complete by now. In the vSphere Client select **webserver03** and power it on (or reboot it if it's already powered on).



Task 4.2 - Verify Access to RDS from webserver03

1. In the vCenter browser tab , Select **webserver03**
2. Review and record **webserver03 IP address**
3. Click **LAUNCH WEB CONSOLE**
4. In the browser tab for webserver03. Log in as
5. login: **root**
6. password: **VMware1!**

You'll need this IP when creating a NAT rule for webserver03

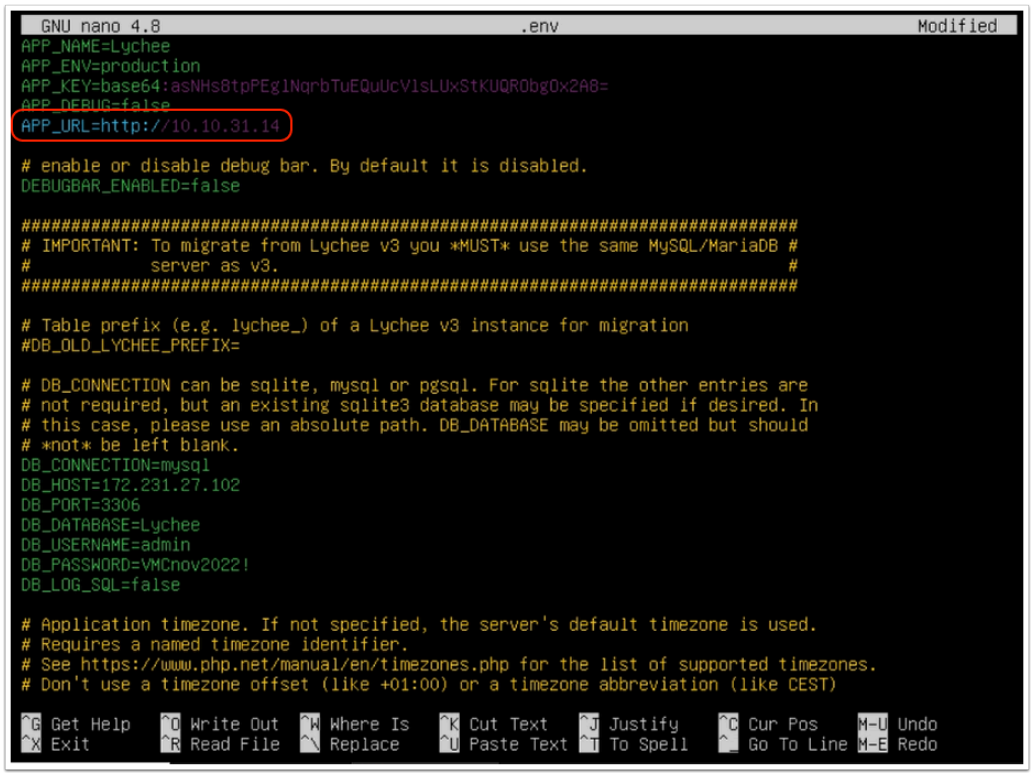


6. At the shell prompt enter the following commands

```
cd /var/www/html/Lychee
nano .env
```

Click to copy

7. Replace the **APP_URL** IP with your webserver03 IP (Note: You recorded this IP in the steps above)
8. Press **CTRL+O** and **Enter** to save the change
9. Press **CTRL+X** to close the file

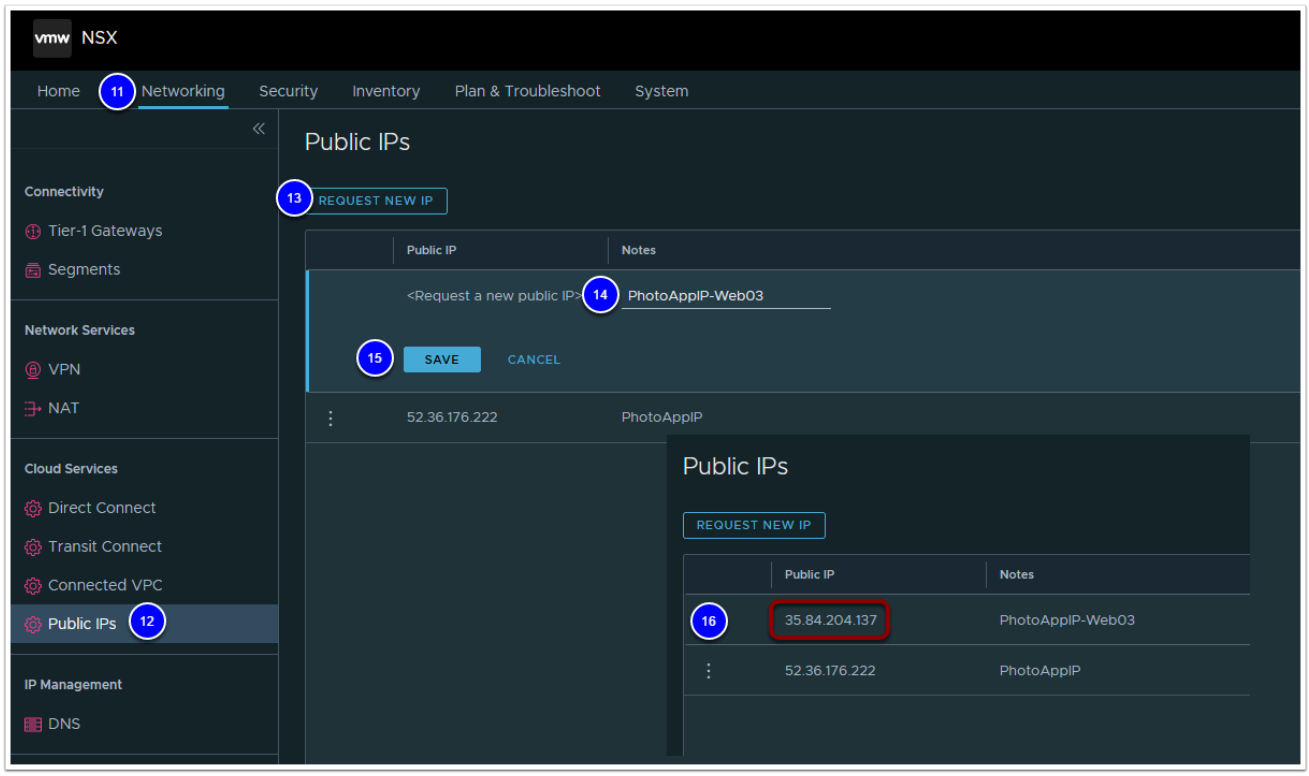


10. Reboot **webserver03** by typing the following command

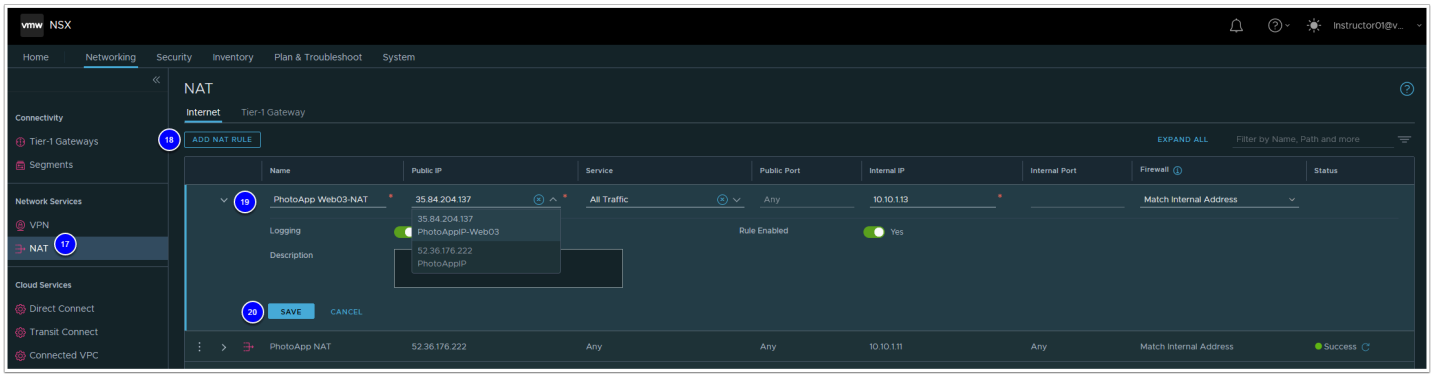
```
reboot
```

Click to copy

11. In the NSX Manager UI browser tab, click **Networking** tab
12. Click **Public IPs**
13. Click **Request New IP**
14. Type **PhotoAppIP-Web03** in the Notes field
15. Click **Save**
16. Record the IP generated, you will use it to configure NAT for webserver03 and access the application



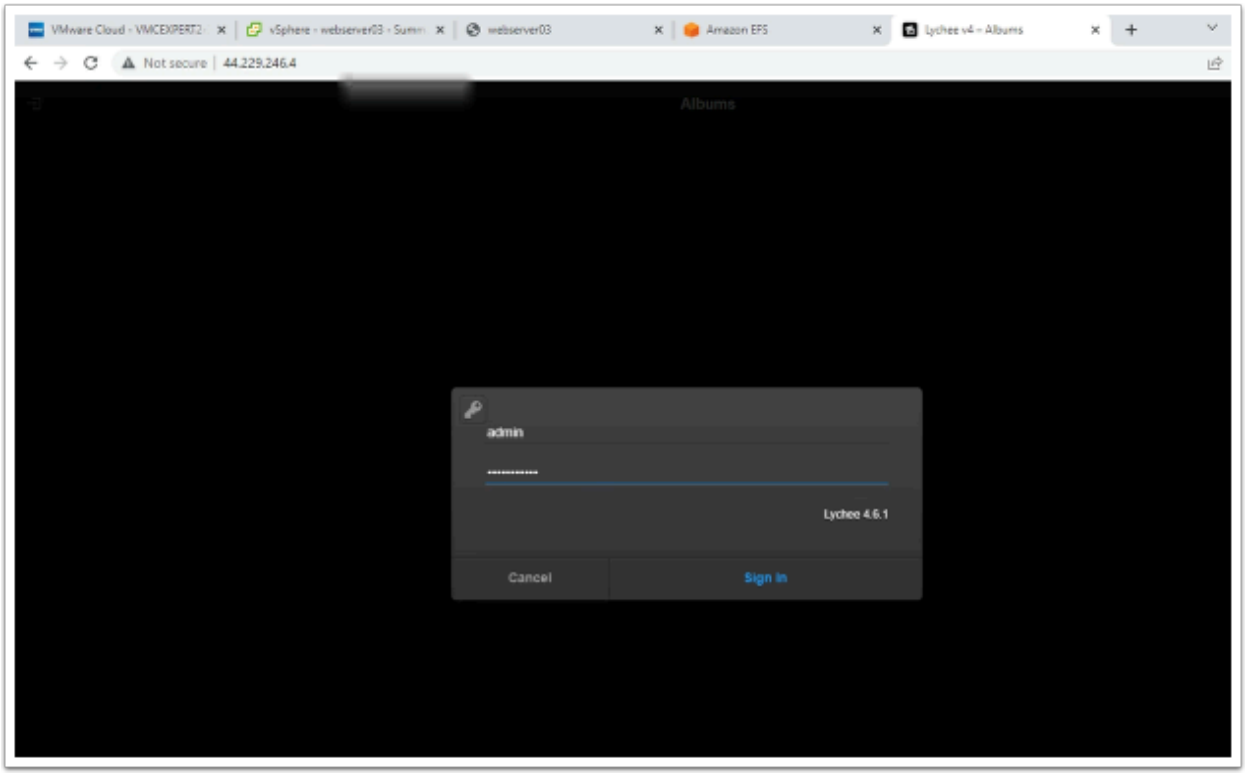
17. Click **NAT**
18. Click **Add NAT Rule**
19. Configure the rule as follows:
    - Name: **PhotoApp Web03-NAT**
    - Public IP: **{The public IP you generated and store in step 16)**
    - Internal IP: **{The IP address o}f webserver03}**
    - Logging: **Enabled**
20. Click **Save**



With webserver03 rebooted, we will now confirm it's connectivity to the RDS MySQL DB. We want to confirm webserver03 can reach the DB and you are able to log in.

21. In a new browser tab, Type in the **{Public IP (NAT IP)}** for **webserver03** (Step 16)
22. At the Login screen log in as:
    - **Username: admin**
    - **Password: {AWS Console Password provided by your instructor}**
23. Click **Sign In**

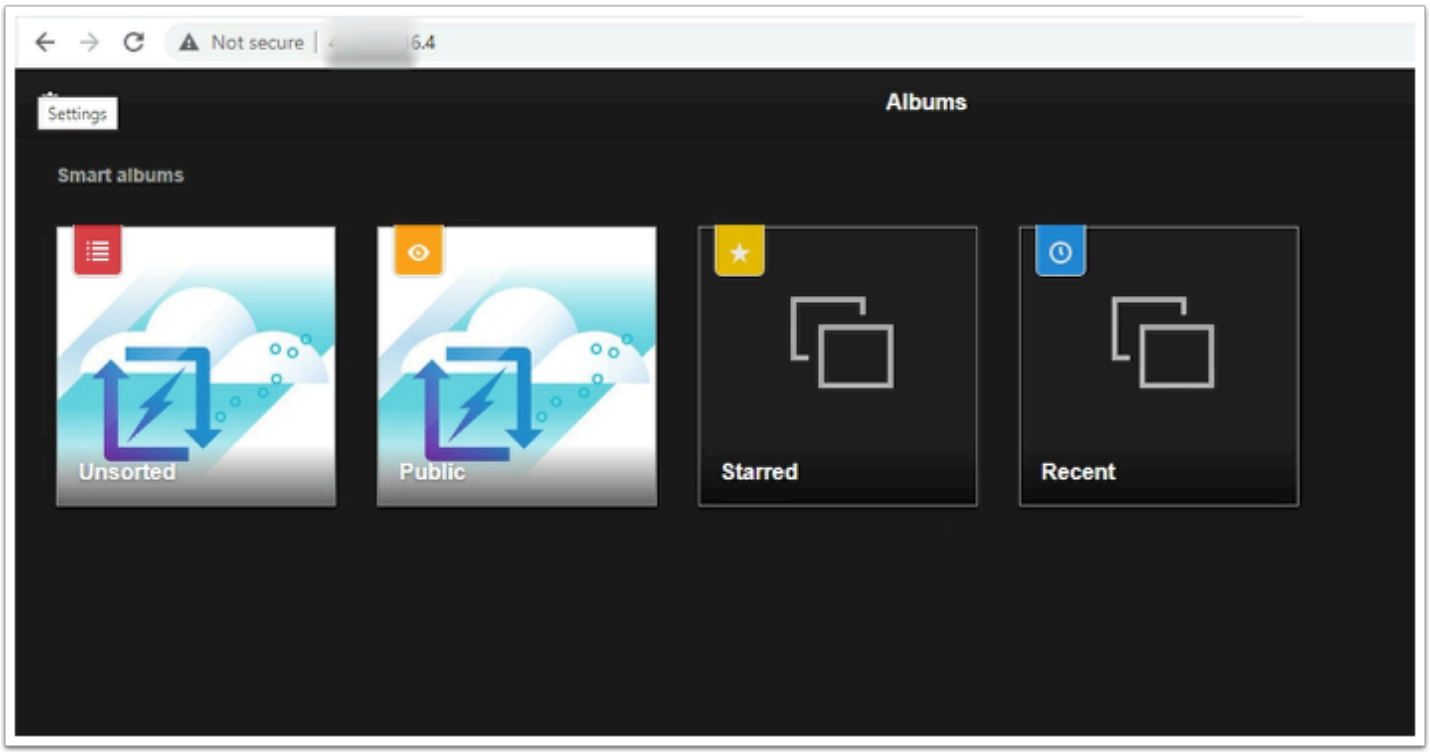
> ⚠ NOTE: This is the same account you created when you configured webserver01 against the RDS DB in Task 3.3, steps 20 & 21. If you created an account other than instructed then use that account instead.



24. Click the Public folder, or any folder you previously uploaded an image to

Questions

1. Why didn't you have to run the Lychee Configuration wizard on webserver03 as you did on webserver01?
2. Are the images you uploaded from webserver01 visible from webserver03?
3. If you were to upload new images on Webserver03 will they be visible on webserver01?

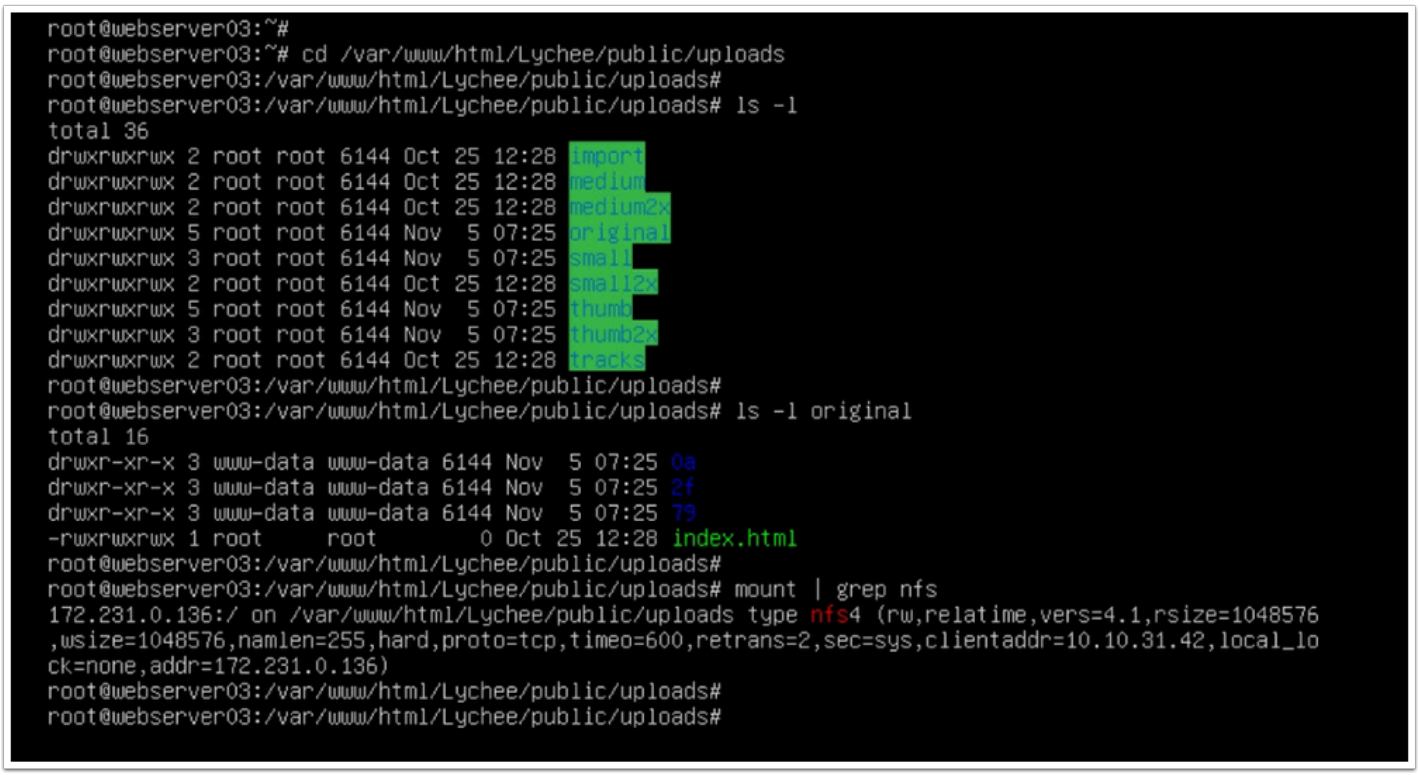


Task 4.3 - Confirm EFS Mount on Webserver03

1. In the **webserver03** browser tab, If needed, log in to **webserver03** as:
   - **user: root**
   - **password: VMware1!** (If the browser tab was previously closed, go to vCenter and open console to webserver03 and continue)
2. Type the following commands to confirm NFS share is mounted to **webserver03**:

```
cd /var/www/html/Lychee/public/uploads
ls -l
ls -l original
mount | grep nfs
```

Click to copy



Additional Lab 2 - Load-balancing Applications in VMC on AWS with Amazon Application Load balancer

In this lab, we will show how to leverage an Amazon Application Load Balancer (ALB) with Virtual Machines running in a VMware Cloud on AWS SDDC.

In this session we will load balance webserver01 and webserver03 (PhotoAppVM). We will then test connectivity to the PhotoVMApp via the Amazon Application Load-Balancer.

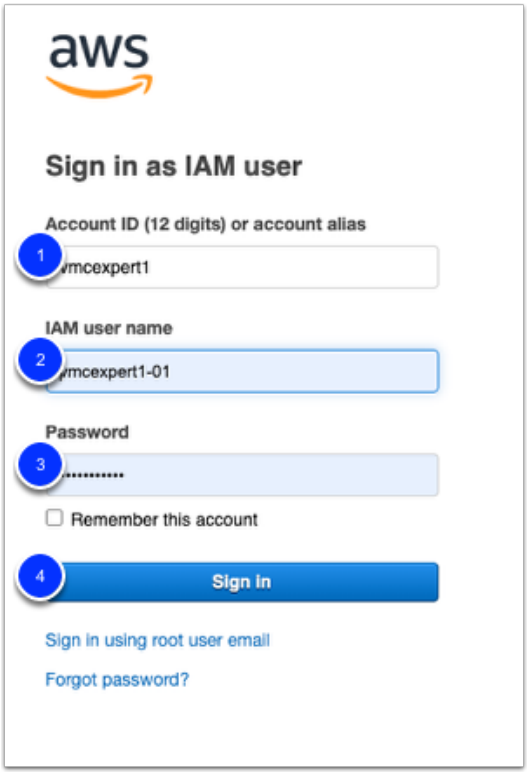
We will begin by requesting a public IP for webserver03 and define a NAT rule for it. Doing so ensure webserver03 is addressable from the internet and not just the private application network.

Task 1 - Add Web Servers to the Amazon Application Load Balancer

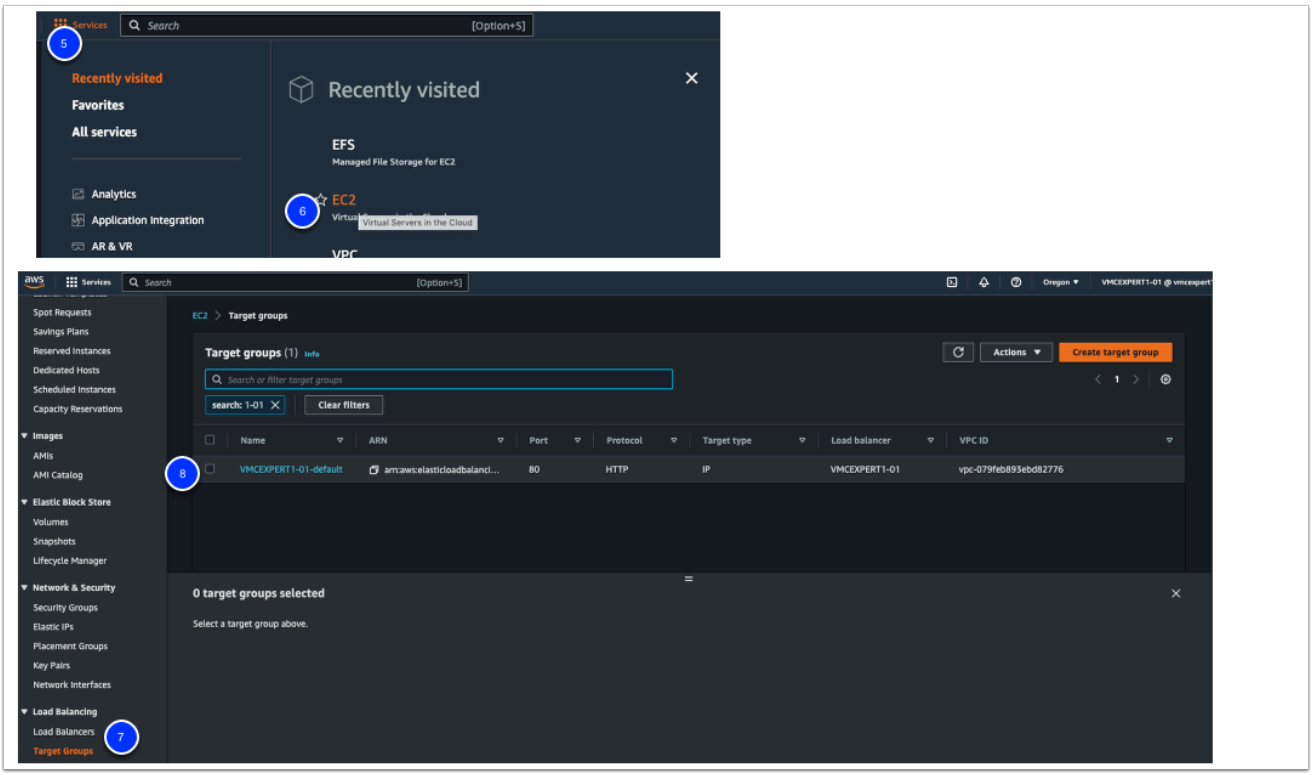
On your browser, open a new tab and go to: https://vmcexpert{#}.signin.aws.amazon.com/console where {**#**} indicates your AWS environment (1, 2 or 3)

The Credentials below are from the AWS Console portion of your student lab assignment sheet

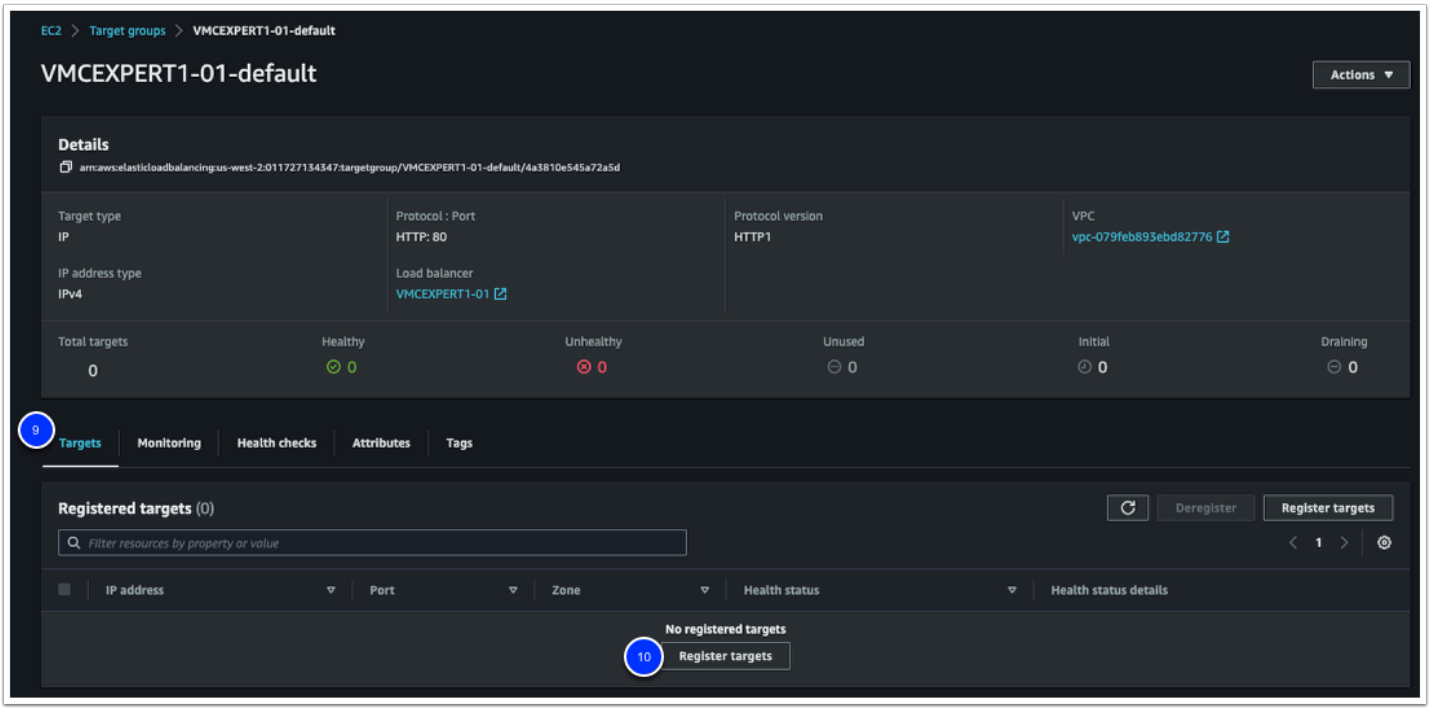
1. Account ID or alias: **vmcexpert# i.e vmcexpert1, vmcexpert2 or vmcexpert3**
2. IAM user name: **VMCEXPERT#-XX**(where **#** is your Environment ID and **XX** is the number assigned to you)
3. Password: <**AWS Console PW provided By your instructor**>
4. Click **Sign In**



5. In the upper left-hand Click **Services**
6. Under Recently visited click **EC2**
7. In the Left pane under **Load Balancing**, Click **Target Groups** (you may have to scroll down)
8. Find and blue click the text for your target group <**vmcexpert#-xx-default**> (where **XX** is your student number)

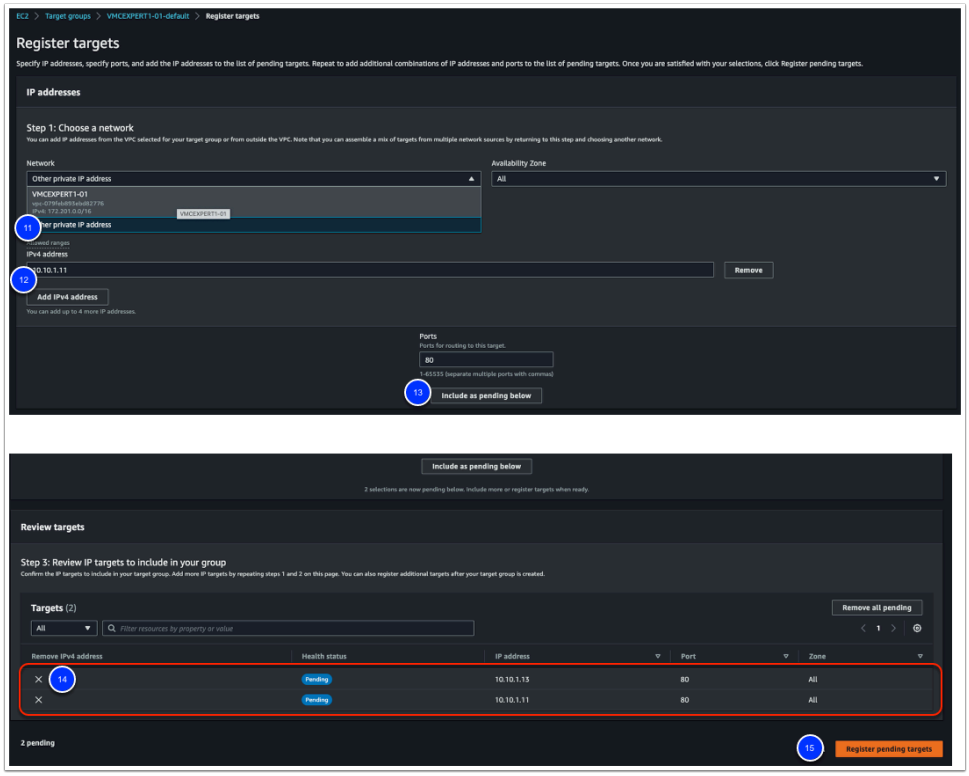


9. Click the **Targets** tab
10. Click **Register targets**



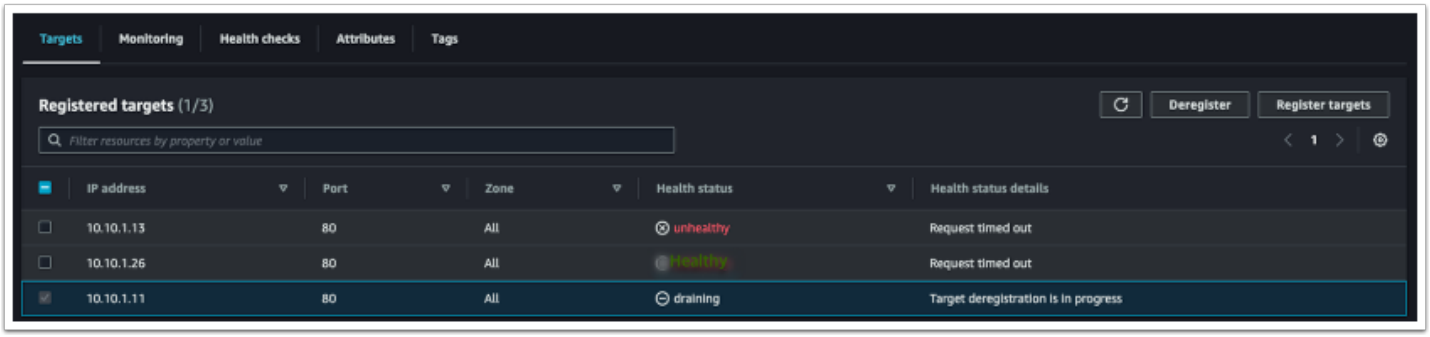
11. In the Network Drop-Down list, select **Other Private IP address**
12. In the IP Field Enter the <**Private IP address of Webserver01**>
13. Click **Include as pending below**

14. Repeat steps 12 & 13, this time using the <**Private IP for webserver03**>
15. Click **Register pending targets**



10. Wait 10-20 seconds, click the refresh circle, the status should turn from '**initial**' to '**healthy**'

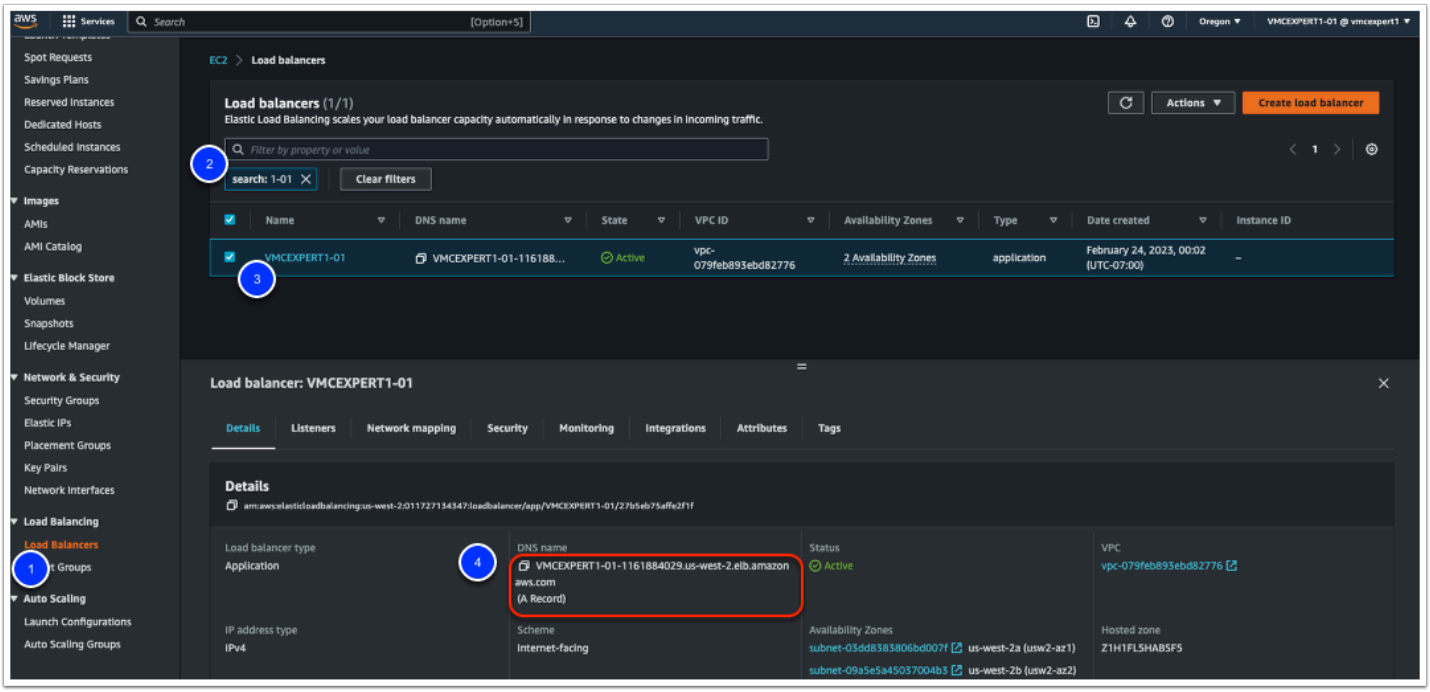
> **Webserver03** should display a **Healthy** state, but **webserver01** should be **Unhealthy**. This is expected because webserver01 is powered-off. Keep it powered-off for now.



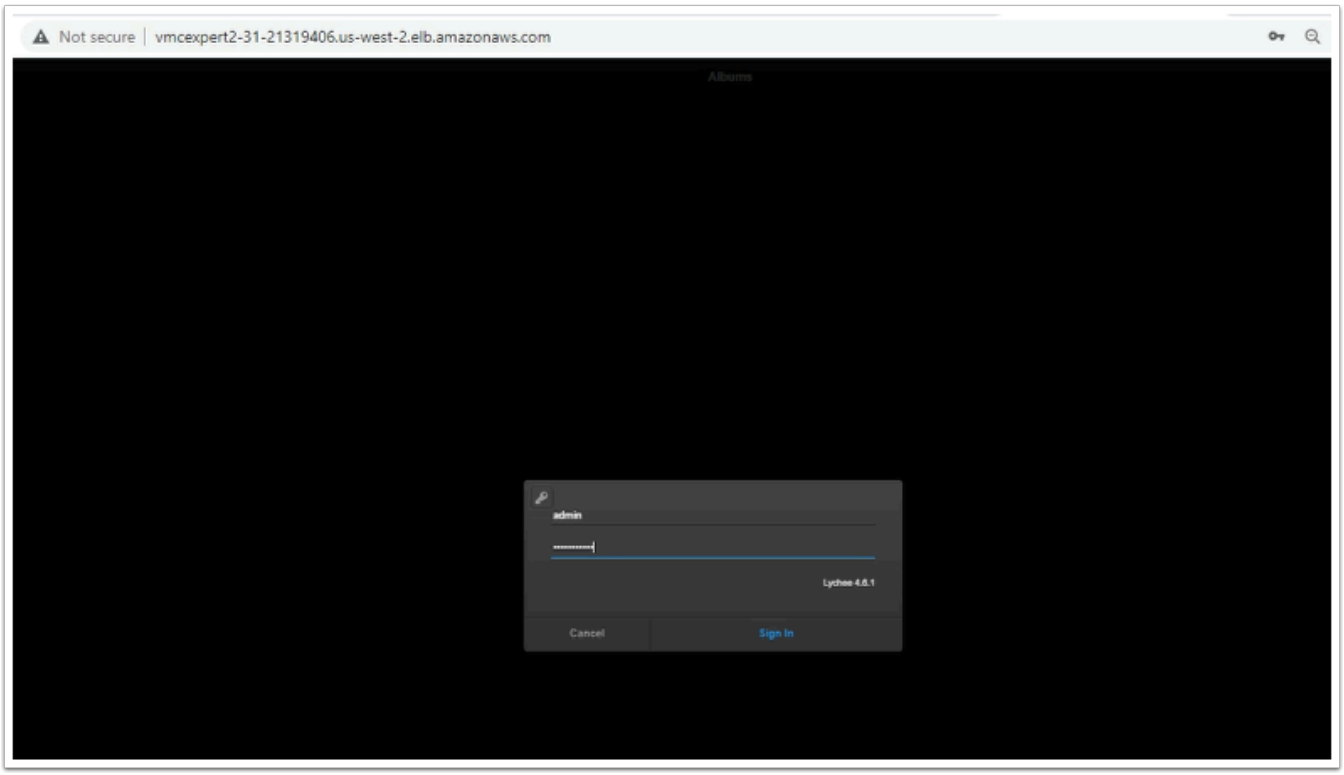
Task 2 - Validate the Application Load Balancing

1. In the menu on the left, under **Load Balancing**, click **Load Balancers**
2. Type <**VMCEXPERT#-XX**> in the Search field to Find your load balance, Where **XX** = your student number. i.e. **VMCEXPERT3-01**
3. Check the box next to your Load Balancer (don't click on the blue text)

4. From the Description tab copy the DNS name ie. VMCEXPERT#-XX-UID.(region).elb.amazonaws.com



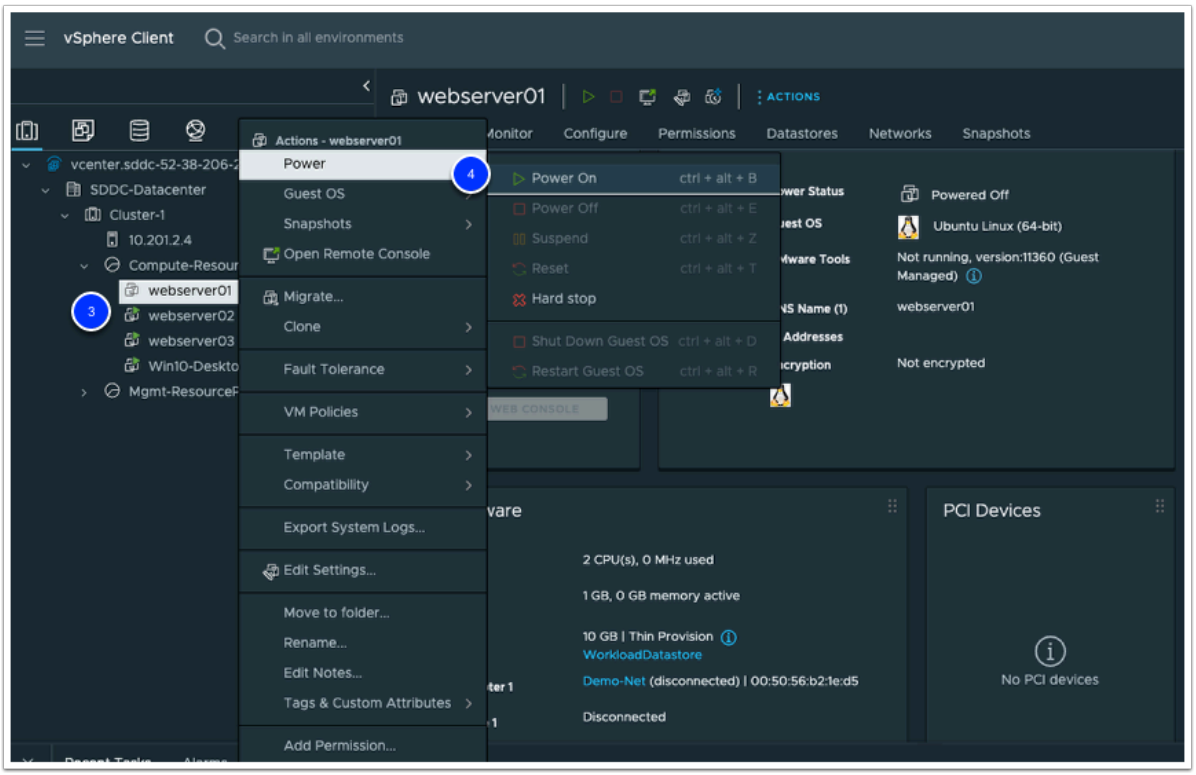
4. Paste the **DNS Name** in your browser to access the PhotoApp via ALB i.e. **vmcexpert3-01-888644610.eu-central-1.elb.amazonaws.com/Lychee**
5. If you aren't prompted for a login, Click the exit icon in the upper-right hand of the application page
6. When prompted to login in use the following**:**
   - Username: **admin**
   - Password: **<Password Provided by your instructor>**
   - Click **Sign In**
7. Upload some additional images to **webserver03**



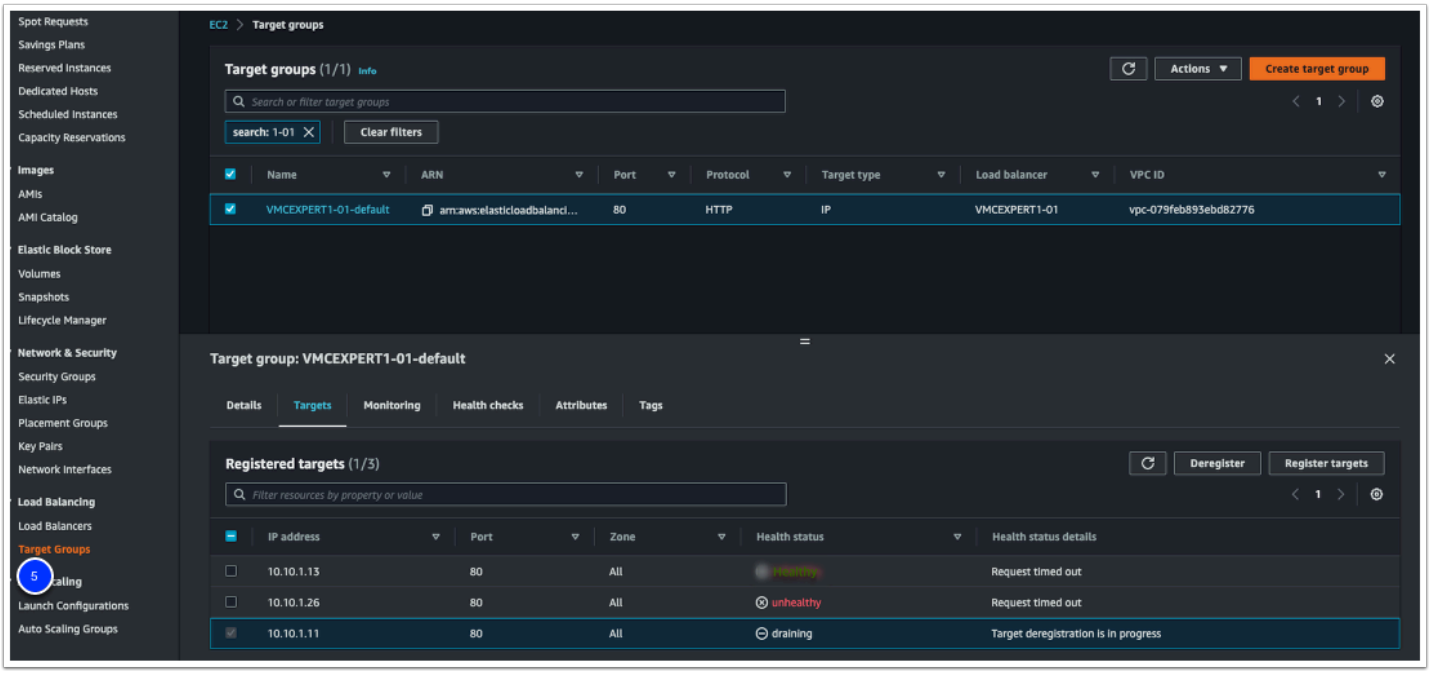
Task 3 - Test Load Balancer functionality

1. In your vCenter browser tab, select **webserver01**
2. Right-click **webserver01**,
3. Click **Power --> Power-on**
4. Right-click **webserver03**
5. Click **Power--> power-off**
6. Confirm that **webserver01**'s IP address has not changed 10.10.x.11

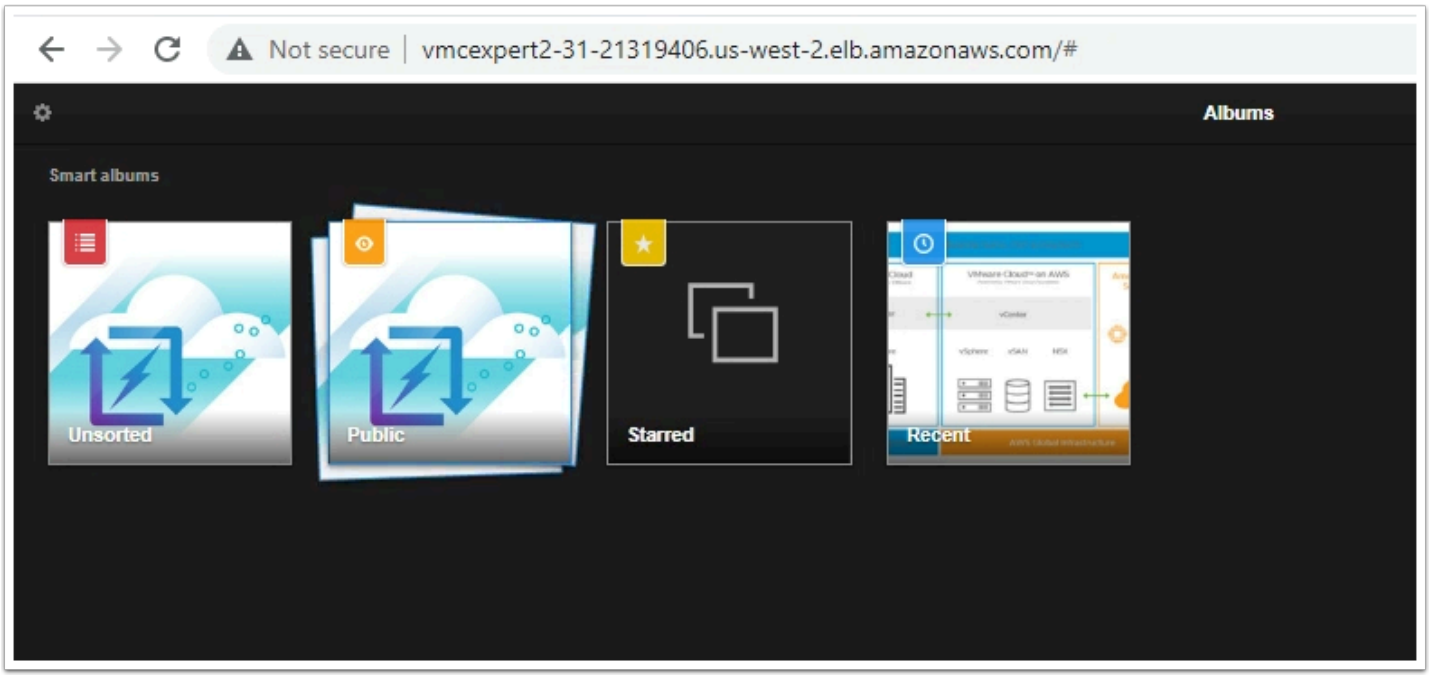
NOTE: This was the IP we used when we configured the Load Balancer Target Group



7. In the AWS Console, under **Load Balancing** select **Target Groups**
8. Select your <**VMCEXPERT#-XX>-default** (Your Load Balancer Target Group) where XX is your student number
9. Click the **Targets** tab
10. After 60 secs the powered off VM state should report unhealthy



10. In a new Google Chrome incognito window type <**your ALB DNSName**> i.e. vmcexpert3-01-1218955224.eu-central-1.elb.amazonaws.com
11. If prompted to login in use the following**:**
    - Username: **admin**
    - Password: **<Password Provided by your instructor>**
    - Click **Sign In**
12. Power-on your previously powered-off vm in step 2



Conclusion

A separate software load balancer is not required to be deployed in the VMware stack to provide load-balancing functionality for your Applications running in VMware Cloud on AWS. There is no additional updating or maintenance to be performed with your load balancer as you can instead use the one provided by AWS.