VMware Cloud on AWS Workshop Lab Guide

**GETTING HANDS ON WITH VMWARE CLOUD ON AWS AND NATIVE AWS SERVICE INTEGRATIONS**

ENT306-R / ENT306-R1

# Welcome

Welcome to the VMware Cloud on AWS workshop. This lab guide is also hosted in S3 for your convenience.

<https://vmc-workshop-20191111220357408200000004.s3-us-west-2.amazonaws.com/common/re-Invent_ENT306_Combined_Lab_Guide_v1.pdf>

Your login will in the format

studentNNN

where NNN will be the place-card number on your seat (e.g student034). Please receive your password from the workshop organizer.

You can complete all the workshop steps and modules through a browser based virtual desktop interface (VDI).

For an enhanced experience, you can use VMware Horizon7 desktop client to access your virtual desktop. Desktop clients are available for Windows, Mac, Linux, iOS, Android and Chromebook

<https://my.vmware.com/en/web/vmware/info/slug/desktop_end_user_computing/vmware_horizon_clients/5_0>

Table of Contents

[Welcome 1](#_Toc25167017)

[Start here – Login into your virtual desktop 4](#_Toc25167018)

[AWS console login 4](#_Toc25167019)

[VMC console login 4](#_Toc25167020)

[Verify your VMware VMs status 5](#_Toc25167021)

[Module - Security 5](#_Toc25167022)

[Customer Problem 5](#_Toc25167023)

[Proposed Solution 6](#_Toc25167024)

[Create an Application Load Balancer (ALB) Instance 6](#_Toc25167025)

[Log into the web application and verify security settings 7](#_Toc25167026)

[Test exploit the web application 7](#_Toc25167027)

[Cross-Site Scripting (XSS) 7](#_Toc25167028)

[Scenario 8](#_Toc25167029)

[Configure AWS WAF 8](#_Toc25167030)

[Verify WAF Protection 10](#_Toc25167031)

[Module - Monitoring with AWS CloudWatch 11](#_Toc25167032)

[Setup and Architecture 11](#_Toc25167033)

[Lab Steps 12](#_Toc25167034)

[1. Access the workload VM server 12](#_Toc25167035)

[2. Explore the setup/environment 12](#_Toc25167036)

[3. Start workload simulation 14](#_Toc25167037)

[4. Start CloudWatch agent 15](#_Toc25167038)

[5. Build a Monitoring Solution in CloudWatch 16](#_Toc25167039)

[Create a Dashboard 17](#_Toc25167040)

[Conclusion 18](#_Toc25167041)

[Module – Data Resiliency 19](#_Toc25167042)

[Customer Problem 19](#_Toc25167043)

[Proposed Solution 19](#_Toc25167044)

[Architecture Diagram 20](#_Toc25167045)

[Environmental overview 21](#_Toc25167046)

[VMware Cloud on AWS Account 21](#_Toc25167047)

[Linked AWS Account 23](#_Toc25167048)

[Lab instructions 24](#_Toc25167049)

[Configure networking for the AWS Storage Gateway VM 24](#_Toc25167050)

[Register the AWS Storage Gateway 29](#_Toc25167051)

[Set the Storage Gateway local console password 36](#_Toc25167052)

[Create a Storage Gateway File Share 37](#_Toc25167053)

[Mount your file share 39](#_Toc25167054)

[Conclusion 42](#_Toc25167055)

[Module - Analytics 43](#_Toc25167056)

[Customer problem 43](#_Toc25167057)

[Proposed solution 43](#_Toc25167058)

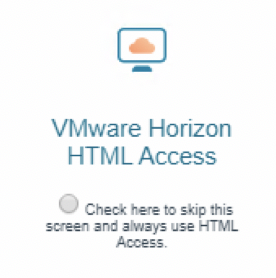
[Thank You 61](#_Toc25167059)

# Start here – Login into your virtual desktop

Launch your browser and go to the url

<https://vdi.vmcworkshop.io/>

Click on VMware Horizon HTML Access



Accept the notice and proceed to login page.

Enter studentNNN login and password. Your virtual desktop will be launched. It takes about 2-3 minutes to boot your Windows desktop.

All the required lab software are available under Lab Shortcuts folder in the desktop.

## AWS console login

In your virtual desktop, open a browser and go to the url

<https://vmc-workshop.signin.aws.amazon.com/console>

Username: studentNNN

Password: [As received from workshop organizer]

Please check if the region is set to us-west-2.

## VMC console login

In your virtual desktop, open a browser and go to the url

<https://vcenter.sddc-52-43-109-233.vmwarevmc.com/ui>

Username: studentNNN

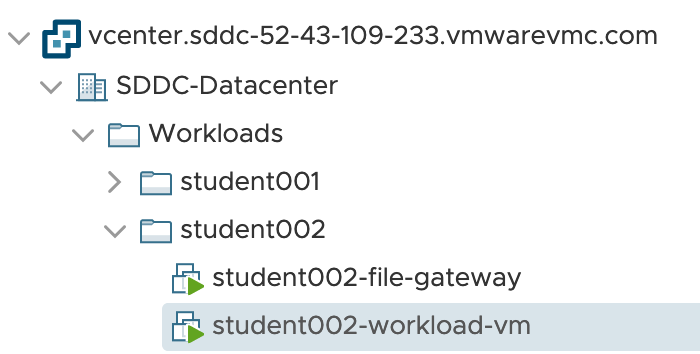
Password: [As received from workshop organizer]

## Verify your VMware VMs status

Login into VMC console as studentNNN user.

Click on “VMs and Templates”

On the next screen, expand “SDDC-Datacenter”, expand “Workloads” folder, expand “studentNNN” folder (where NNN is your allocated number).



There should be 2 VMs up and running with green ‘play’ triangle icon before their names

studentNNN-file-gateway

studentNNN-workload-vm

These VMs will be used throughout the workshop modules.

# Module - Security

In this module, you will learn how to leverage the native AWS security services Amazon Web Application Firewall (WAF) and Amazon Macie to protect VMware Cloud on AWS workloads.

## Customer Problem

Your CIO has tasked you with migrating an on-premises web application to VMware Cloud on AWS and protect the workload according to a set of requirements.

You have then been given the following requirements/guidance from your CIO:

* The solution must protect the workload against various web exploits and vulnerabilities.
* The solution must be highly available and scalable, performant, and offer custom policy configuration.
* The solution should be a managed service.

## Proposed Solution

After some extensive research, you have determined that the Amazon Web Application Firewall with Application Load Balancer will address all the requirements.

## Create an Application Load Balancer (ALB) Instance

1. Log in to the AWS console.
2. Click Services > EC2 > Load Balancing > Load Balancers
3. Click Create Load Balancer.
4. Click Create under the **Application Load Balancer** heading.
5. Step 1: Basic Configuration
   1. Name: alb-studentNNN-workshop (where NNN equals your student identifier)
   2. Scheme: Internet-facing
   3. IP Address Type: ipv4
   4. Listeners (do not add any additional listeners)
   5. Availability Zones
      1. VPC: vpc-098819d02d8b7c324 | vpc-workshop
      2. Availability Zones: us-west-2a, us-west-2b
      3. Click Next: Configure Security Settings
6. Step 2: Configure Security Settings
   1. Ignore the message, “Improve your load balancer's security. Your load balancer is not using any secure listener.”
   2. Click Next: Configure Security Groups
7. Step 3: Configure Security Groups
   1. Assign a security group:
      1. Choose Select an existing security group.
      2. Select the following segurity group: sg-0d6285b666c9793c4 (Name: sg\_internet\_http\_https)
   2. Click Next: Configure routing
8. Step 4: Configure Routing
   1. Target Group
      1. Target group: New target group
      2. Name: tg-studentNNN-workshop (where NNN equals your student identifier)
      3. Target type: IP
      4. Protocol: HTTP
      5. Port: 80
      6. Health checks
         1. Protocol: HTTP
         2. Path: /login.php
   2. Click Next: Register Targets
9. Step 5: Register Targets
   1. Network: Other private IP address.
   2. Availability Zone: all
   3. IP: 10.200.NNN.10 (where NNN equals your student identifier) [This is the IP address of your VMware VM studentNNN-workload-vm running in VMware Cloud on AWS]
   4. Port: 80
   5. Click Add to list.
   6. Click Next: Review
10. Click Create.

## Log into the web application and verify security settings

1. Log in to the AWS console.
2. Click Services > EC2 > Load Balancing > Load Balancers
3. Locate and click on your Load Balancer instance.
4. In the Description output below, locate the DNS name for the load balancer instance (e.g. [alb-student001-workshop-1733099543.us-west-2.elb.amazonaws.com](http://alb-student001-workshop-ent306-1733099543.us-west-2.elb.amazonaws.com))
5. Navigate to the following URL:  
   http://<dns\_name\_from\_step\_4>/DVWA-master/login.php (case-sensitive)
6. Log into the web application portal using the following credentials:
   1. Username: admin
   2. Password: password
7. In the left navigation pane, click DVWA Security.
8. Verify that the security level is set to Low. If necessary, change the setting to Low.
9. Click Submit.

## Test exploit the web application

### Cross-Site Scripting (XSS)

* Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications.
* XSS enables attackers to inject client-side scripts into web pages viewed by other users.
* A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.
* In addition, the attacker can send inputs (e.g., username, password, session ID, etc) which can be later captured by an external script.
* The victim's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.

Follow the scenarios below to gain a better understanding of this type of vulnerability in the context of a web application.

### Scenario

1. Click XSS (Stored) on the left navigation pane.
2. Name: Test 1
3. Type the following in the Message field:

|  |
| --- |
| <script>alert("This is a XSS Exploit!")</script> |

1. Click Sign Guestbook.
2. Click DVWA Security on the left navigation pane.
3. Click XSS (Stored) on the left navigation pane. Notice that the JavaScript alert we just created is now displayed. Every Time a user comes to this forum, this XSS exploit will be displayed. This exploit can be easily modified to capture cookie/session information for future man-in-the-middle attacks.

## Configure AWS WAF

Follow the steps below to configure the Amazon WAF service to mitigate one of the top web application attack categories - Cross-Site Scripting.

Steps:

1. Navigate to the WAF & Shield service in the AWS Console.
2. Click Conditions > Cross-site Scripting
3. Select US-WEST (Oregon) in the Region filter.
4. Click Create Condition.
   1. Name: condition-studentNNN-owasp-a3-xss (where NNN equals your student identifier)
   2. Region: US-WEST (Oregon)
   3. Enter the following filters:
      1. Part of the request to filter on: Query String
      2. Transformation: URL decode
      3. Click Add filter.
      4. Part of the request to filter on: Query String
      5. Transformation: HTML decode
      6. Click Add filter.
      7. Part of the request to filter on: URI
      8. Transformation: URL decode
      9. Click Add filter.
      10. Part of the request to filter on: URI
      11. Transformation: HTML decode
      12. Click Add filter.
      13. Part of the request to filter on: Body
      14. Transformation: URL decode
      15. Click Add filter.
      16. Part of the request to filter on: Body
      17. Transformation: HTML decode
      18. Click Add filter.
      19. Part of the request to filter on: Header
      20. Header\*: Cookie
      21. Transformation: URL decode
      22. Click Add filter.
      23. Part of the request to filter on: Header
      24. Header\*: Cookie
      25. Transformation: HTML decode
      26. Click Add filter.
   4. Click Create.
5. Click AWS WAF > Rules.
6. Click Create rule.
   1. Name: rule-studentNNN-mitigate-xss (where NNN equals your student identifier)
   2. CloudWatch metric name: rulestudentNNNmitigatexss (where NNN equals your student identifier)
   3. Rule Type: Regular
   4. Region: US-WEST (Oregon)
   5. Under Add conditions section, select the following condition syntax:  
      When a request does match at least one of the filters in the cross-site scripting match condition (select condition-studentNNN-owasp-a3-xss from the drop-down list) (where NNN equals your student identifier)
   6. Click Create.
7. Click AWS WAF > Web ACLs.
8. Click Create web ACL.
   1. Name: webacl-studentNNN-owasp (where NNN equals your student identifier)
   2. CloudWatch metric name: webaclstudentNNNowasp (where NNN equals your student identifier)
   3. Rule Type: Regular
   4. Region: US-WEST (Oregon)
   5. Resource type to associate with web ACL: Application load balancer
   6. AWS resource to associate: alb-studentNNN-workshop-ent306 (where NNN equals your student identifier)
   7. Click Next.
   8. Click Next.
   9. Under Add rules to web ACL:
      1. Select Block as the Action.
      2. Select rule-studentNNN-mitigate-xss from the drop-down list. Click Add rule to web ACL.
      3. Select Block as the Action.
      4. Under Default action\*, select Allow all requests that don’t match any rules.
   10. Click Review and create.
   11. Once satisfied with the configuration, click Confirm and create.

## Verify WAF Protection

Follow the steps below to repeat the cross-site scripting attempt on the web application:

1. Click XSS (Stored) on the left navigation pane.
2. Click OK on the pop-up dialog
3. Name: Test 2
4. Type the following in the Message field:

|  |
| --- |
| <script>alert("Hello XSS!")</script> |

1. Click Sign Guestbook.
2. The attempt to exploit the application this time should result in a HTTP 403 - Forbidden message to the end-user. This indicates a successful block of the attempt by the AWS WAF service.

# Module - Monitoring with AWS CloudWatch

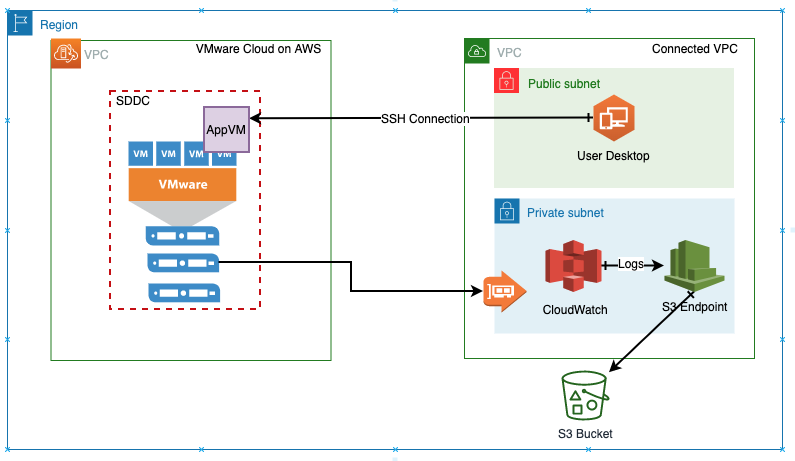
**VMware Cloud on AWS** provides private connectivity between your VMware SDDC and your native AWS account. This allows VMs residing inside your VMware Cloud SDDC to consume native AWS services securely using the same simple and flexible pay-as-you-go model as you would do within your native AWS environment.

In this lab module we will learn how take advantage of the private connectivity between VMware SDDC and native AWS account to add advanced monitoring capabilities to your VMware workloads running on VMware Cloud on AWS. Here are the key learning objectives:

* VMware Cloud on AWS connectivity model
* Basics of Amazon CloudWatch Service
* Common use-case for system and application monitoring
* Options for advanced monitoring using AWS and Partner services

## Setup and Architecture

The key concept here is that with VMware Cloud on AWS you are able to create a private VPC endpoint for CloudWatch service inside your “connected VPC[[1]](#footnote-1) so when a VMware VM attempts to log CloudWatch event the data goes over the cross-account ENI link instead of public Internet. The architecture diagram below highlights that



## Lab Steps

We will start by logging into our test VM, that has all the software pre-requisites already installed. We’ll just to quick walk through the environment and once we make sure everything is in order, we will use a script to simulate a web traffic hitting the web server installed locally on the VM. As the web server processes the incoming request, it will also log the activity to the local server log files. We will configure the CloudWatch agent to send these logs over to CloudWatch service to populate monitoring dashboard. In addition to monitoring the log files generated by the web server, we will also capture several key metrics to monitor the VM’s overall health and performance characteristics. Lets us reiterate the fact that all the communication between our test VM residing on VMware Cloud on AWS SDDC and the CloudWatch backend is taking over a private ENI based connection, keeping your data secure, so let’s get started.

## Access the workload VM server

In your virtual desktop, open the Lab Shortcuts folder

Click “Putty (64-bit)” shortcut link

In the Putty window, under Hostname (or IP address) section

Type in your workload VM hostname

studentNNN-workload-vm.sddc.vmcworkshop.io

Leave the port as 22, and click “Open” button on bottom right of Putty window screen.

Click “Yes” if you are prompted for server host key caching

When prompted for “login as:”, type in vmc-user

Password: [As received from the workshop organizer]

## Explore the setup/environment

Verify that your AWS CLI is installed and credentials and profiles have been configured.

Enter following command in you ssh window

aws cloudwatch list-dashboards --profile AmazonCloudWatchAgent --region us-west-2

You should see some output similar to the one shown below. If you get an error message <ASK\_FOR\_HELP\_TEXT>

{

"DashboardEntries": [

{

"DashboardName": "subnet-036c6a2b2fa30a8f3\_i-04082bbe1134379b0",

"LastModified": "2019-11-12T08:21:52Z",

"DashboardArn": "arn:aws:cloudwatch::229790159028:dashboard/subnet-036c6a2b2fa30a8f3\_i-04082bbe1134379b0",

"Size": 7188

},

]

}

Verify that CloudWatch agent is installed and configured

In order to save time in this lab, the CloudWatch agent has already been installed and configured on your system. You can verify that the agent has been installed by running the following command

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status

You should see the agent installed but in currently in the “stopped” state:

*<Boatner; Please change this so if the agent is already running don’t try to restart>*

{

"status": "running",

"starttime": "",

"version": "1.231221.0"

}

The configuration file for CloudWatch agent can be found at this location:

/opt/aws/amazon-cloudwatch-agent/bin/config.json

Please open this file in your favorite text editor, vi, nano etc, and examine its content.

Make sure to use “sudo” before the command, for example,

sudo cat /opt/aws/amazon-cloudwatch-agent/bin/config.json

Do you see a section that looks like this?

"collect\_list": [

{

"file\_path": "/etc/httpd/logs/access\_log",

"log\_group\_name": "student-000",

"log\_stream\_name": "{hostname}"

}

Lets replace student-010 with your actual student-NNN log group name.

If you are student002 then the command would be

sudo sed -i ‘s/student-010/student-002/g’ /opt/aws/amazon-cloudwatch-agent/bin/config.json’

Verify web server and test script

*Apache web server is already installed and configured to run your system. You can verify its status by running the following the command*

systemctl status httpd

It should result in the following output, if the web server is not running, or you get an error, please <ASK\_FOR\_HELP\_TEXT>

httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)

Active: active (running) since Wed 2019-11-13 22:51:23 UTC; 7h ago

Docs: man:httpd.service(8)

Main PID: 35787 (httpd)

Status: "Total requests: 2; Idle/Busy workers 100/0;Requests/sec: 7.24e-05; Bytes served/sec: 0 B/sec"

CGroup: /system.slice/httpd.service

For this module, we will use a script to generate some traffic to your web server, the script can be found in the home directory of the current user

/home/vmc-user/sim.py

## Start workload simulation

Now that we have verified that we have all the basic components of the lab environment, we are ready to start our simulation. Please run the workload simulation script mentioned in the previous section by running the following command.

python sim.py

Your initial output on the screen will look like this

Starting workload simulation...

HTTP Status Code: 404

HTTP Status Code: 200

HTTP Status Code: 200

.

.

.

The script sends both valid and invalid requests to the web server. Each request results in a log entry being made web servers **access\_log** file. You can keep the script running in this window and open a new SSH session and run the following command

sudo tail -f /etc/httpd/logs/access\_log

Do you see log entries being added? They should look like:

2.7.16 Linux/4.14.146-120.181.amzn2.x86\_64"

127.0.0.1 - - [14/Nov/2019:07:32:54 +0000] "GET /index.html HTTP/1.1" 200 20 "-" "python-requests/2.6.0 CPython/2.7.16 Linux/4.14.146-120.181.amzn2.x86\_64"

127.0.0.1 - - [14/Nov/2019:07:32:54 +0000] "GET /index.html HTTP/1.1" 200 20 "-" "python-requests/2.6.0 CPython/2.7.16 Linux/4.14.146-120.181.amzn2.x86\_64"

127.0.0.1 - - [14/Nov/2019:07:32:54 +0000] "GET /index.html HTTP/1.1" 200 20 "-" "python-requests/2.6.0 CPython/2.7.16 Linux/4.14.146-120.181.amzn2.x86\_64"

You can exit the “**tail**” command using “**Ctrl-C**” Or simply let it run in the current window and start a new SSH window to launch CloudWatch agent in

## Start CloudWatch agent

Now that our simulation is generating some log data, let’s start the Amazon CloudWatch agent that will send log data to CloudWatch for further monitoring. Enter the following command

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m onPremise -c file:config.json -s

You’ll see some message scroll by on the screen. To confirm that CloudWatch agent is up and running use the following command

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status

Output:

{

"status": "running",

"starttime": "2019-11-14T07:42:13+0000",

"version": "1.231221.0"

}

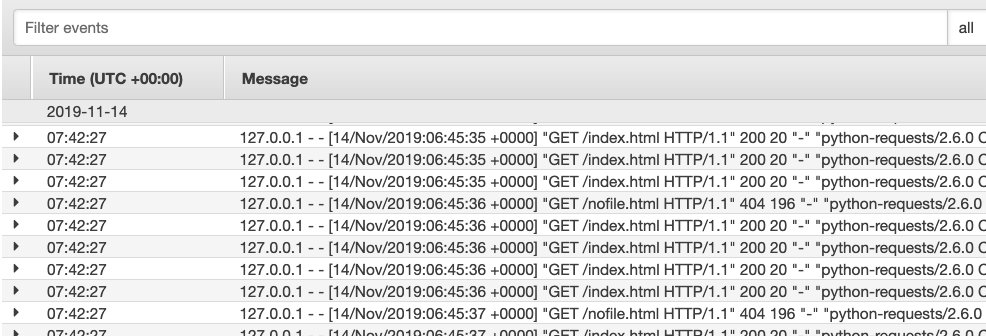
## Build a Monitoring Solution in CloudWatch

Open a browser tab and login to the CloudWatch service in AWS Console. You can paste the URL below in your browser

<https://us-west-2.console.aws.amazon.com/cloudwatch/home?region=us-west-2#dashboards:>

Verify that if log data is being sent over from your VM to CloudWatch:

1. In the navigation sidebar, click “Logs”
2. In the “**Filter**” box, enter your student-id, for example “student-001” (case sensitive)
3. Under “**Log Groups”** click on your name matching your student-id and then click on the entry matching your VMs hostname under “**Log Streams**”. Do you see data similar to what is shown below? This confirms that CloudWatch agent is sending logs from your VM running on VMware Cloud on AWS SDDC to CloudWatch

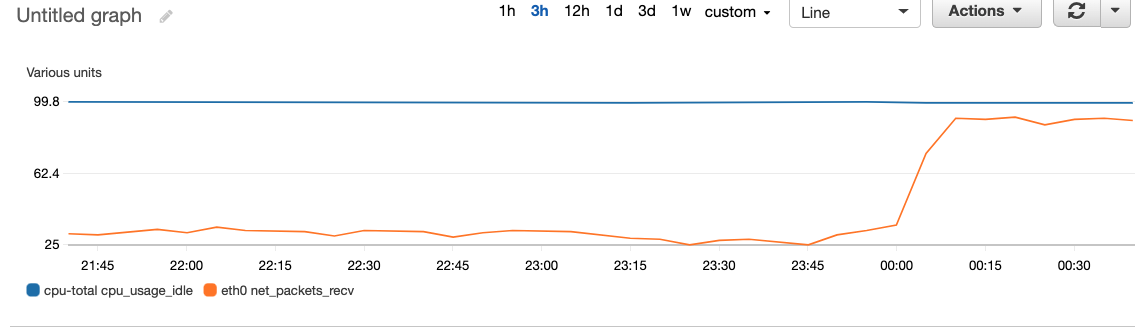


Verify if the “Metrics” are being sent from VM by the CloudWatch agent:

1. Now click on “Metrics” in the sidebar
2. Do you see “CWAgent” under “Custom Namespaces” on the right-hand side? Click on it



1. Select **cpu, host** and “**cpu\_usage\_idle**” under “**Metric Name**”. You should be able to see the graph showing some data.



CloudWatch can capture several key metrics and performance indicators from your VMs running in VMware Cloud. Feel free to explore other metrics in this section as well.

### Create a Dashboard

Now you can start building a custom dashboard to display the most critical health and performance indicators in a single pane of glass view.

1. Click “Dashboards” in the left-hand navigation bar
2. Click “Create Dashboard”
3. Enter the name for your Dashboard as your full student-id. Please use caution here, only access and work on your own dashboard
4. First add a “Text widget” to give your dashboard a name and purpose. If you are familiar with Markdown formatting please feel free to customize, otherwise you can just paste the sample code

# Module 2: Monitoring

## Student-XXX

-----

### VMware Cloud on AWS

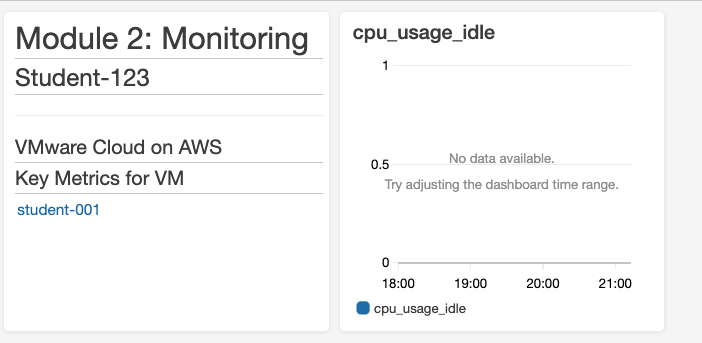
### Key Metrics for VM

[student-001](#dashboards:name=student-XXX)

1. After adding the text widget, we can add a widget showing Idle CPU time. Click “Add Widget” 🡪 select “Line” 🡪 click “Configure” 🡪 CWAgent 🡪 “**cpu, host**” 🡪 select checkbox for “**Metric Name: cpu\_usage\_idle**” making sure that you select the host that matches your assigned student-id and hostname, for example, if your assigned name is “localhost”



1. Once you have completed the steps, your final dashboard will look similar to the one shown below



## Conclusion

Given the short amount of time we have to complete this lab, we have featured only some the very basic capabilities that CloudWatch provides to help you effectively monitor your VMs running on VMware Cloud on AWS. Some additional ideas to consider may include

* Single Dashboard for application and system level monitoring
* Automating log cycling to save on storage cost
* Defining custom metrics and publish them to a common dashboard.

In addition to CloudWatch, VMware Cloud on AWS also supports integration with other monitoring solutions developed by our partners include Dynatrace, New Relic to name a few.

# Module – Data Resiliency

In this module, you will get hands‑on experience with integrating AWS storage services into your VMware Cloud on AWS software‑defined data center (SDDC) to learn cost effective ways to protect your valuable data.

## Customer Problem

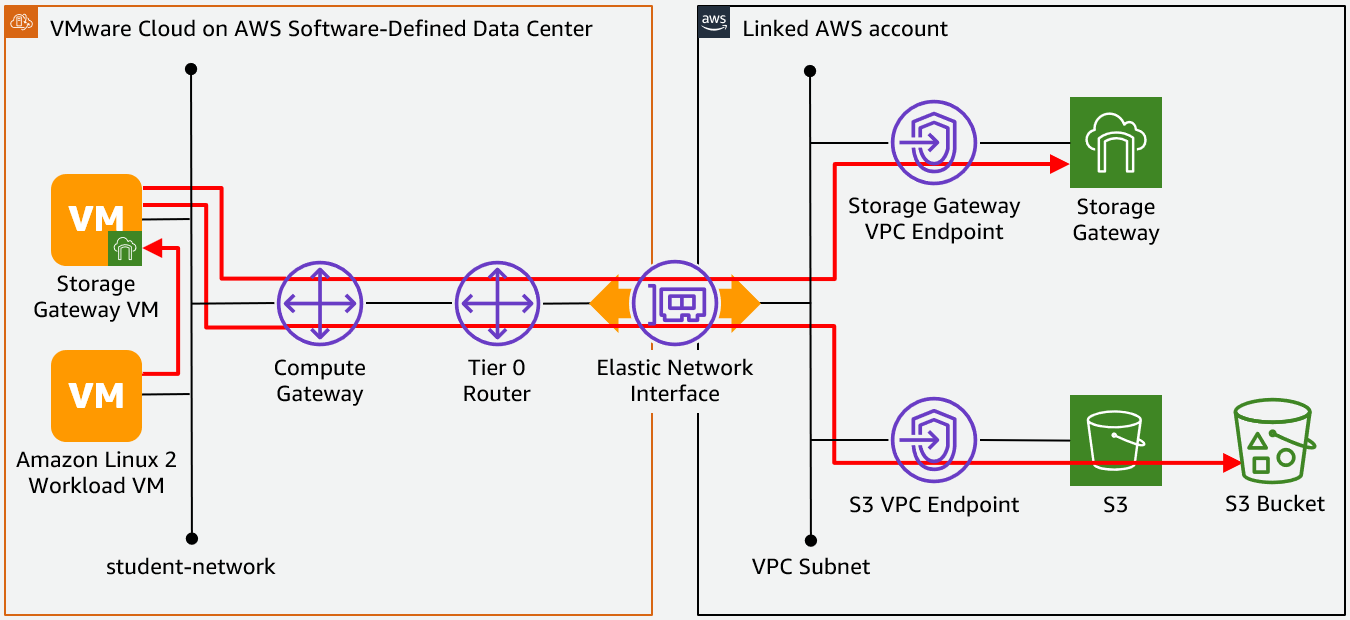
Your CIO has requested that you build a new NFS‑based file storage solution for the company’s existing VMware Cloud on AWS SDDC. The data that will be stored on this solution is mission critical and regulated, so the data must not transit the public internet, must be encrypted in‑transit & at‑rest, and the solution must be highly durable, available, and scalable. The workloads that will consume this solution are primarily read‑intensive and require low‑latency performance. Additionally, bandwidth sent across the VMware Cloud Elastic Network Interface (ENI), which provides the private link between the VMware Cloud on AWS account and the linked AWS account, should be minimized in order to prevent contention with the other bandwidth‑heavy workloads provisioned in the SDDC that share this link.

## Proposed Solution

After some research & experimentation, you have come to learn that the following AWS services can be leveraged to address all of the requirements:

* AWS Storage Gateway: this hybrid cloud storage service can be deployed as a VM in the VMware Cloud on AWS SDDC, and configured as an NFS file server when provisioned in File Gateway mode. Files are cached locally on the File Gateway VM, and stored in an Amazon S3 Bucket. This is optimal for this scenario since the most frequently accessed files would be cached locally on the VM- minimizing latency for the read-intensive workload, and also minimizing the bandwidth required for the VMware Cloud ENI.  
  + Amazon Elastic File System (EFS) was also considered; however, all storage traffic would have to traverse the VMware Cloud ENI and would not be cached locally, so Storage Gateway was deemed a better fit for this solution.
* Amazon S3: is a storage service that is designed to provide high durability and availability, provides native encryption, and offers virtually unlimited storage scalability.
* Amazon VPC Endpoints: enable you to privately connect to AWS services such as S3. The S3 VPC Endpoint is a gateway endpoint type, which works by injecting routes for the S3 network prefixes into the specified routing tables, causing the S3 traffic to be routed to the VPC Endpoint instead of to the public internet. The Storage Gateway VPC Endpoint is an interface endpoint type, which appears as a ENI in each VPC subnet that you specify, and a special fully qualified domain name (FQDN) is generated for resolving the private IP addresses assigned to each VPC Endpoint ENI.

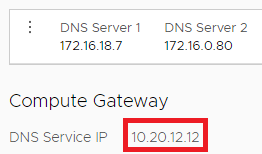
### Architecture Diagram



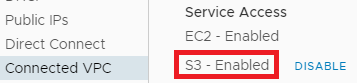
## Environmental overview

|  |
| --- |
| The ‘#’ used below represents your student lab number. For example, if you are assigned student number ‘001’, your Workload VM would be named: ‘student**001**-workload-vm’ and it would be assigned IP address: ’10.200.**1**.10’ |

### VMware Cloud on AWS Account

* vCenter
  + URL: <https://vcenter.sddc-52-43-109-233.vmwarevmc.com/ui/>
  + Username: **WS\student###**
  + Password: **Amazon1!**
* Resource Pool: **student###**
  + VM Folder: **student###**
  + Workload VM
    - VM Name: **student###-workload-vm**
    - IP address: **10.200.#.10**
    - Username: **vmc-user**
    - Password: **Amazon1!**
  + File Gateway VM
    - VM Name: **student###-file-gateway**
    - IP address: Automatically assigned via DHCP
    - Username: **admin**
    - Password: **password**
* DNS Service IP: **10.20.12.12**
* Network Segment
  + Type: **Routed**
  + Name: **student-network**
  + Gateway IP address: **10.200.0.1**
  + Prefix Length: **16**
  + DHCP Range: **10.200.200.0-10.200.255.250**
  + DNS Suffix: **sddc.vmcworkshop.io**

|  |
| --- |
| Each student has been allocated a portion of the ‘student-network’ network segment described above: 10.200.#.0/24, where ‘#’ is your student number. |

* Distributed Firewall Policy
  + Each student’s Amazon Linux 2 Workload VM has been permitted open connectivity to their allocated portion of the ‘student‑network’ network segment as described above and denied connectivity to the rest of the network- including the DHCP range.
* S3 Service Access has been configured to route all traffic destined for the S3 prefixes to the VPC Endpoint Gateway configured in the linked AWS account described below.  
  

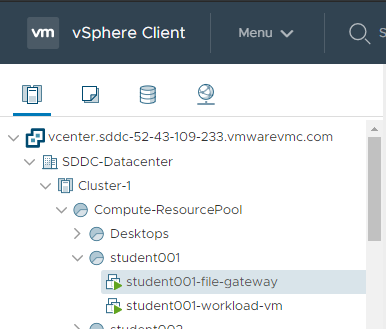
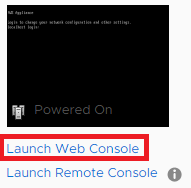
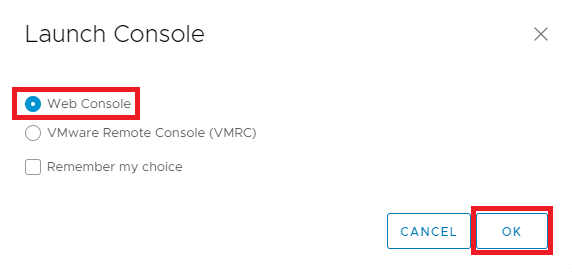
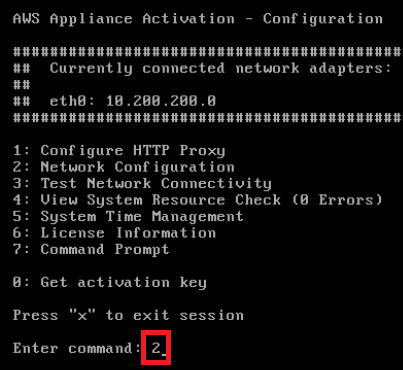
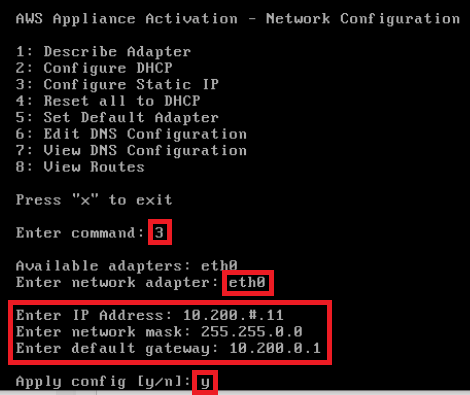
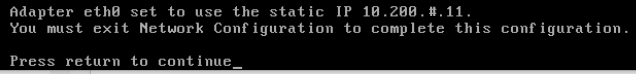
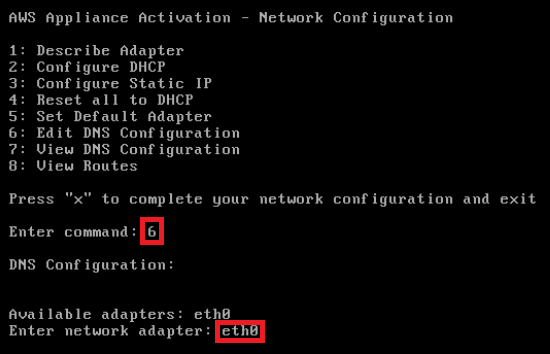
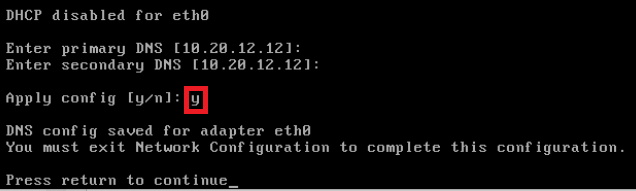
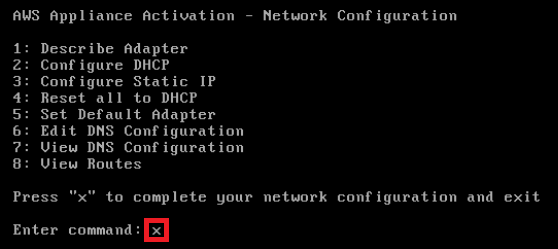
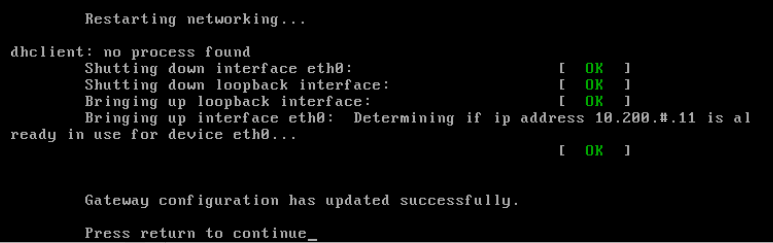
### Linked AWS Account

* AWS Console
  + URL: <https://vmc-workshop.signin.aws.amazon.com/console>
  + Username: **student###**
  + Password: **Amazon1!**
* Storage Gateway VPC Endpoint
  + URL: <https://us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#Endpoints:sort=vpcEndpointId>
  + FQDN: **vpce-0eeced9ed51dc2554-hg28nq7b.storagegateway.us-west-2.vpce.amazonaws.com**
* CloudWatch Log Group
  + URL: <https://us-west-2.console.aws.amazon.com/cloudwatch/home?region=us-west-2#logs:prefix=/aws/storagegateway/student>
  + Name: **/aws/storagegateway/student###**
* S3 Bucket
  + URL: <https://s3.console.aws.amazon.com/s3/buckets/vmc-lab-resiliency-20191111220140298500000003/?region=us-west-2&tab=overview>
  + Name: **vmc-lab-resiliency-20191111220140298500000003**
* IAM Role
  + URL: <https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/AWSStorageGatewayS3BucketAccess>
  + ARN: **arn:aws:iam::229790159028:role/service-role/AWSStorageGatewayS3BucketAccess**

## Lab instructions

### Configure networking for the AWS Storage Gateway VM

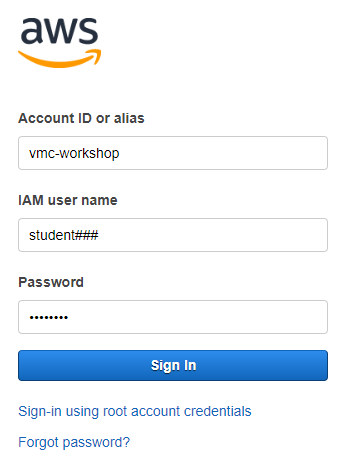
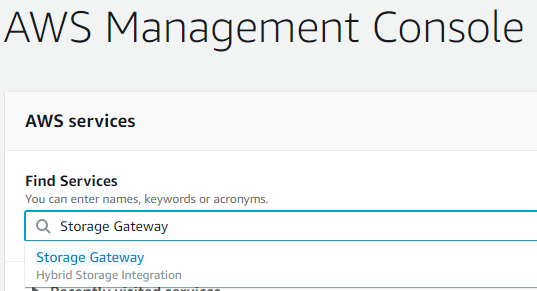
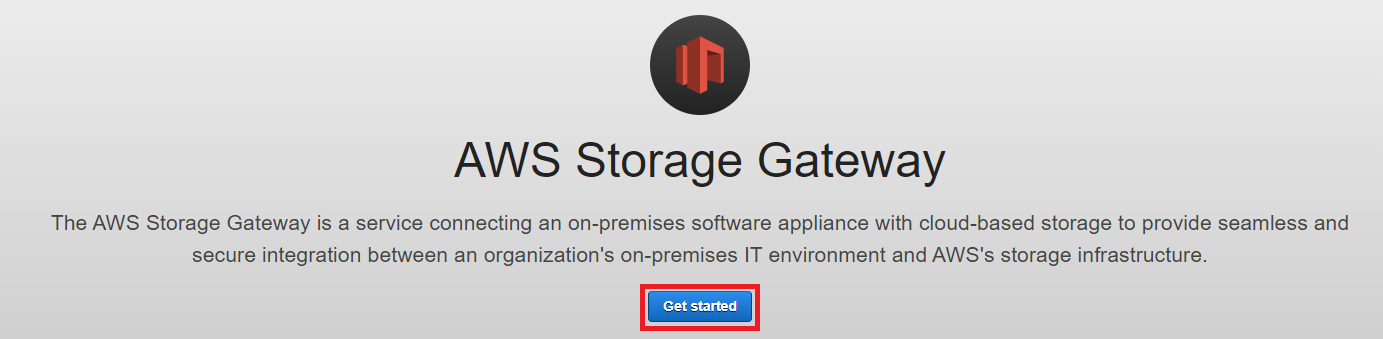
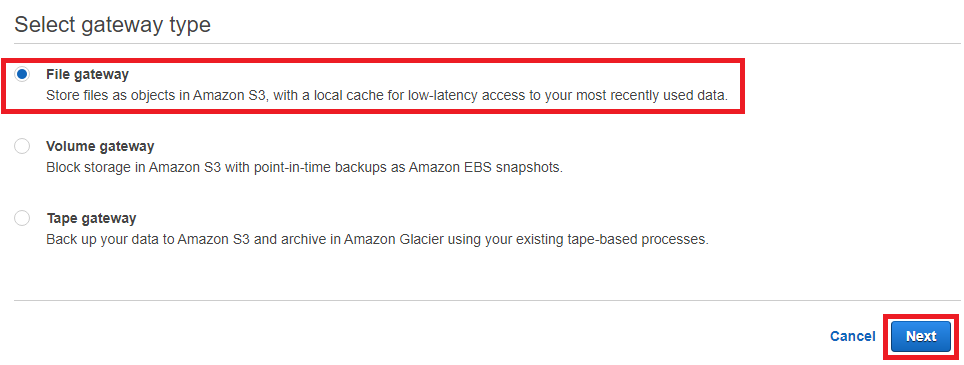
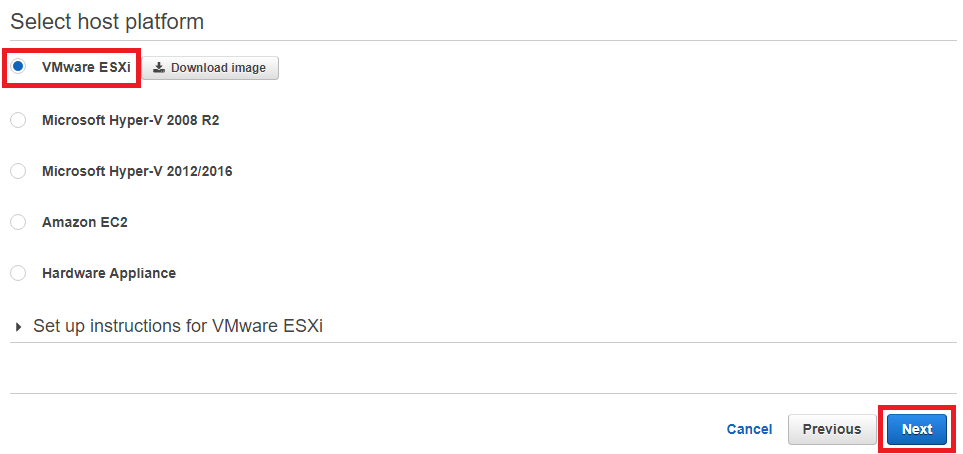
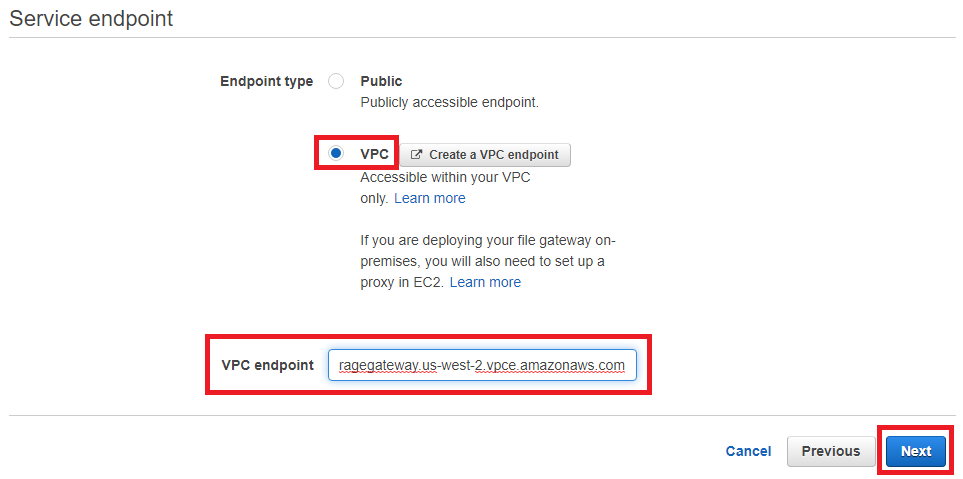
In this section, we will configure a static IP address on the Storage Gateway VM.

1. Log into vCenter
   1. URL: <https://vcenter.sddc-52-43-109-233.vmwarevmc.com/ui/>
   2. Username: **WS\student###**, where ‘#’ is your student number
   3. Password: **Amazon1!**
2. Expand **SDDC-Datacenter** > **Cluster-1** > **Compute-ResourcePool** > **student###**, where ‘#’ is your student number
3. Select the **student###-file-gateway** VM, where ‘#’ is your student number  
   
4. Click **Launch Web Console**  
   
5. Leave the default of **Web Console** and click the **OK** button  
   
6. In the new browser tab, enter the default credentials for the AWS Storage Gateway:
   1. Username: **admin**
   2. Password: **password**
7. From the AWS Appliance Activation – Configuration menu, enter **2** for Network Configuration  
   
8. Enter **3** to configure a Static IP address  
   
   1. Enter **eth0** to specify the network adapter
   2. Enter IP address **10.200.#.11**, where # is your lab number
   3. Enter network mask: **255.255.0.0**
   4. Enter default gateway: **10.200.0.1**
   5. Enter **y** to apply the config
9. Hit enter to continue  
   
10. Back at the main menu, enter 6 to edit the DNS configuration  
    
11. Hit enter twice to accept the default that was configured via DHCP **10.20.12.12**, then **y** to apply the config, and then enter to continue  
    ****
12. Enter **x** to complete the network configuration and exit  
    
13. If successful, you will see: **Gateway configuration update has updated successfully.**, and then hit enter to continue ****

### Register the AWS Storage Gateway

In this section, we will register the AWS Storage Gateway in the linked AWS account and configure it to send the traffic over the private link between the VMware Cloud on AWS account and the linked AWS account.

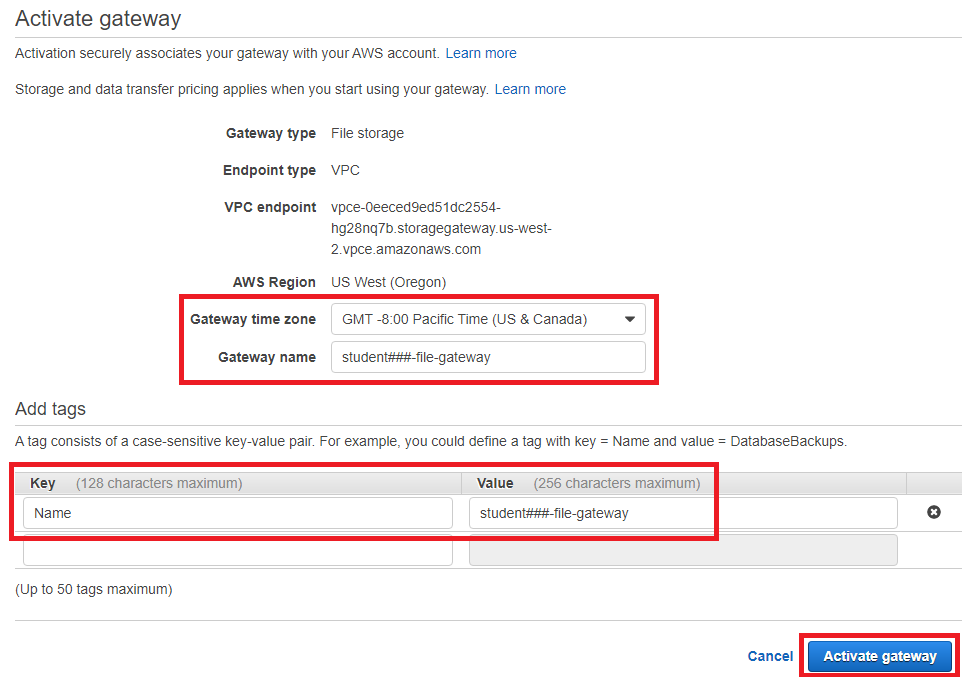
|  |
| --- |
| Direct connectivity to the VM is required, so this section must be completed from the virtual desktop. |

1. Log into the AWS Console from your virtual desktop  
   
   1. URL: <https://vmc-workshop.signin.aws.amazon.com/console>
   2. Username: **student###**, where ‘#’ is your student number
   3. Password: **Amazon1!**
2. In the Find Services search field, enter **Storage Gateway**  
   
   1. URL:<https://us-west-2.console.aws.amazon.com/storagegateway/home/gateways?region=us-west-2>
3. If no other students have activated their storage gateway, you will see the AWS Storage Gateway getting started page, and click the **Get Started** button  
     
     
   Otherwise, you will see the Storage Gateway Console as depicted below, and click the **Create gateway** button  
   
4. Leave the default type of **File Gateway** and then click **Next  
   **
5. Leave the default host platform of **VMware ESXi** (since the Storage Gateway VM was deployed in a VMware Cloud on AWS vSphere cluster), and then click **Next**  
   
6. Select **VPC** for the endpoint type so that we send all Storage Gateway traffic over the private link between the VMware Cloud on AWS account and the linked AWS account. A single Storage Gateway VPC Endpoint has already been provisioned for all students to share for this exercise. Enter the VPC Endpoint FQDN below, and then click **Next**.  
   
   1. VPC Endpoint:  
      **vpce-0eeced9ed51dc2554-hg28nq7b.storagegateway.us-west-2.vpce.amazonaws.com**

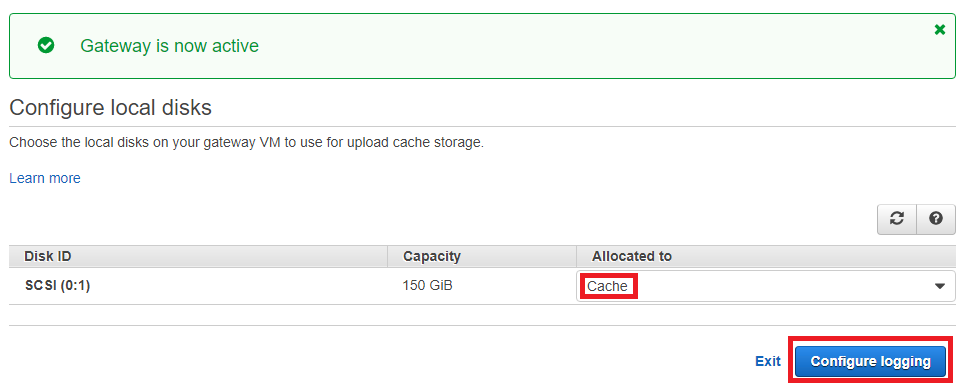
|  |
| --- |
| If you’re interested, check out the Storage Gateway VPC Endpoint in another browser tab: <https://us-west-2.console.aws.amazon.com/vpc/home?region=us-west-2#Endpoints:sort=vpcEndpointId> |

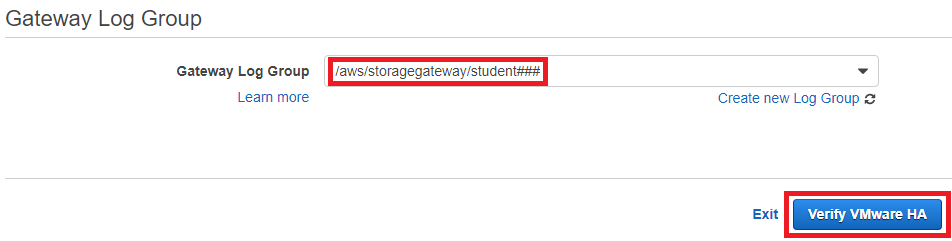
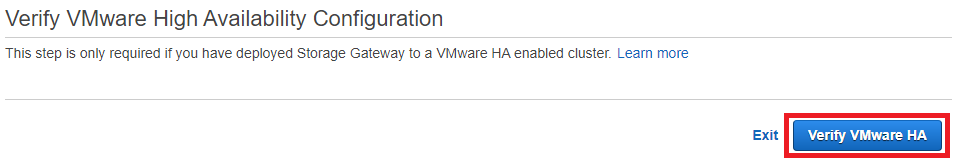
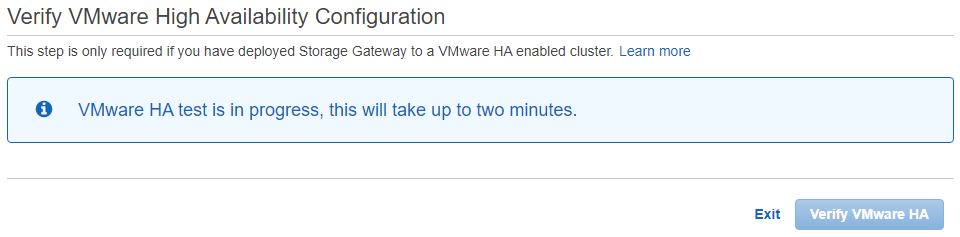
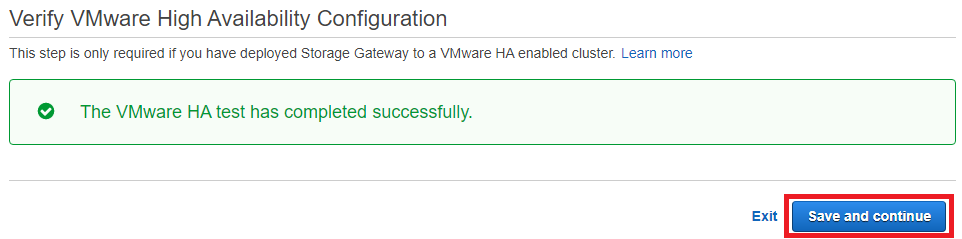
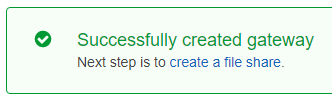
1. Enter the IP address that you configured in the Storage Gateway VM in the previous section  
   
   1. IP address: **10.200.#.11**, where ‘#’ is your student number

|  |
| --- |
| If the AWS Console was not accessed from the virtual desktop, which provides direct connectivity to the Storage Gateway VM, or if the IP address was set incorrectly for the VM during the previous section, this step will fail. |

1. Set the gateway activation configuration  
   
   1. Time zone: **GMT -8:00 Pacific Time (US & Canada)**
   2. Gateway Name: **student###-file-gateway**, where ‘#’ is your student number
   3. Tags
      1. Key: **Name**
      2. Value: **student###-file-gateway**, where ‘#’ is your student number
2. Leave the default of allocating the second virtual hard disk to **Cache**. The cache disk stores recently written files before uploading them to the S3 bucket as well as frequently accessed files for providing faster access to the content.

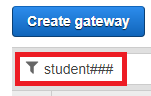
|  |
| --- |
| Occasionally, the secondary virtual disk ID will be detected as ‘SCSI (2:1)’ instead of ‘SCSI (0:1)’. This has not caused issues in testing, so proceed normally. |

****

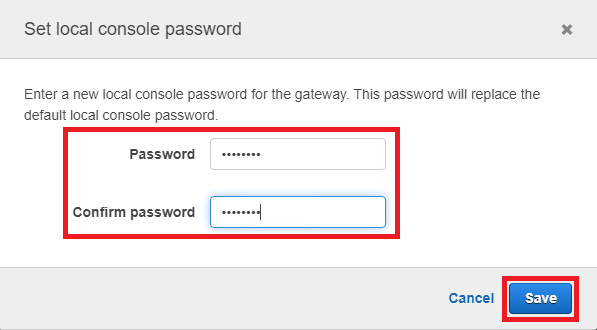
1. Specify the CloudWatch Log Group for storing the Storage Gateway’s event logs and then click **Save and continue**  
   
   1. Gateway Log Group: **/aws/storagegateway/student###**, where ‘#’ is your student number
   2. Click **Verify VMware HA**
2. Click **Verify VMware HA**  
     
     
     
   You will then see the following message:  
     
     
   Followed by the following message confirming that vSphere HA is configured properly (mandatory for VMware Cloud on AWS)  
     
   Click **Save and continue**
3. If all steps were completed successfully, you will see the following message:  
   

### Set the Storage Gateway local console password

Now that we have activated our Storage Gateway, let’s improve our security posture by changing the local console password from the default.

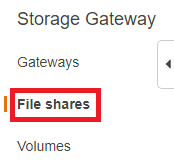
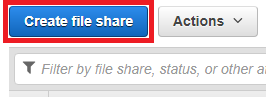
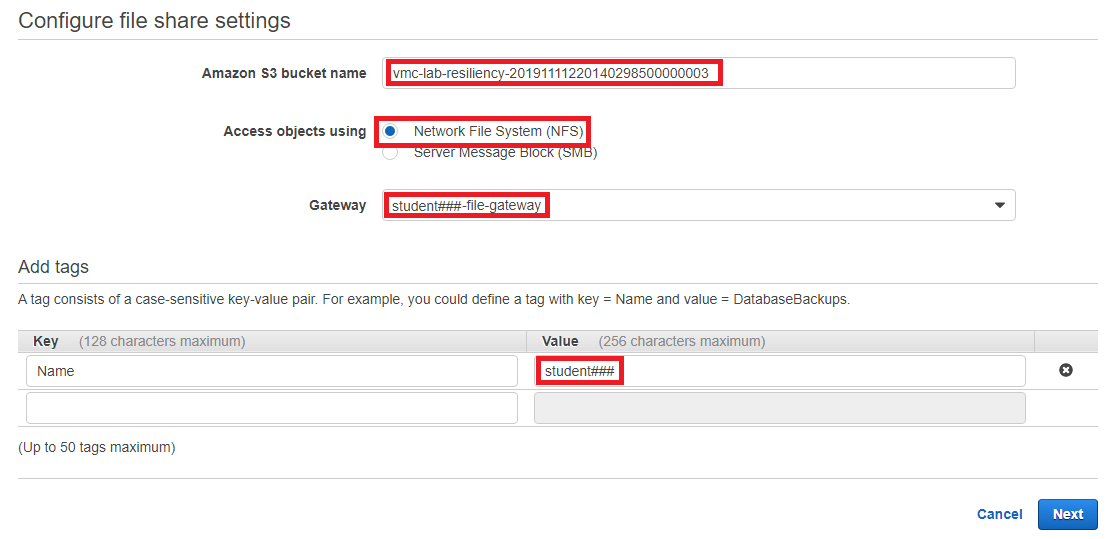
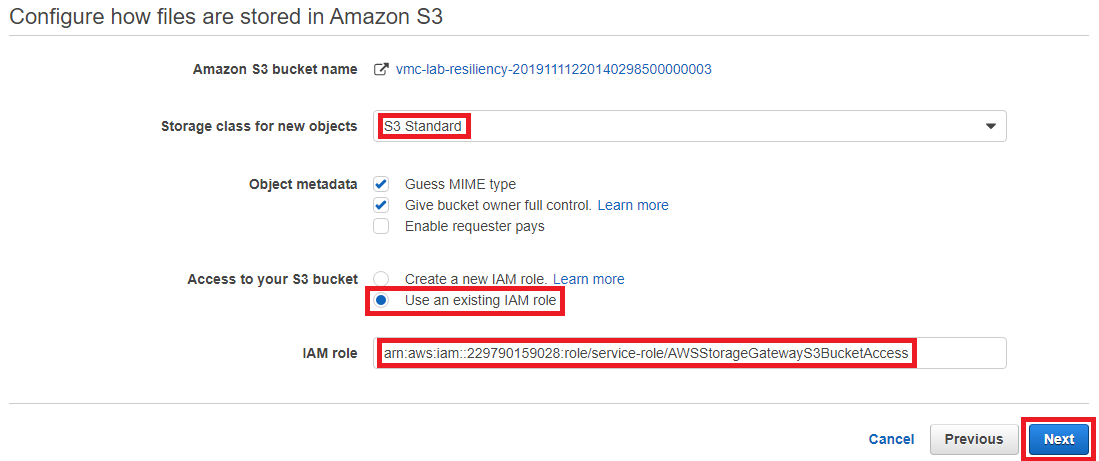
1. Filter for your Storage Gateway, and then select it  
   
2. Click the **Actions** drop down button and then select **Set local console password**  
   
3. Set the password to **Amazon1!**

|  |
| --- |
| This is a weak password that is intended for example use only. Passwords such as this should never be used in a production environment. |



### Create a Storage Gateway File Share

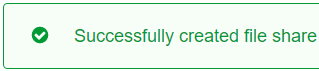
We will now create a Storage Gateway File Share for providing a low‑latency NFS interface for our S3 bucket.

1. In the Storage Gateway Console, select **File shares** from the left-hand menu  
   
   1. URL: <https://us-west-2.console.aws.amazon.com/storagegateway/home/file-shares?region=us-west-2>
2. Click the **Create file share** button  
   
3. Configure the file share settings per the specifications below:  
   
   1. Amazon S3 bucket name: **vmc-lab-resiliency-20191111220140298500000003**
   2. Access objects using: **Network File System (NFS)**
   3. Gateway: **student###-file-gateway**, where ‘#’ is your student number
   4. Tags
      1. Key: **Name**
      2. Value: **student###**, where ‘#’ is your student number
4. Configure how files are stored in Amazon S3  
   
   1. Storage class for new objects: **S3 Standard**

|  |
| --- |
| S3 Standard stores the data in three different Availability Zones within the AWS region, ensuring high availability & durability for your valuable data. |

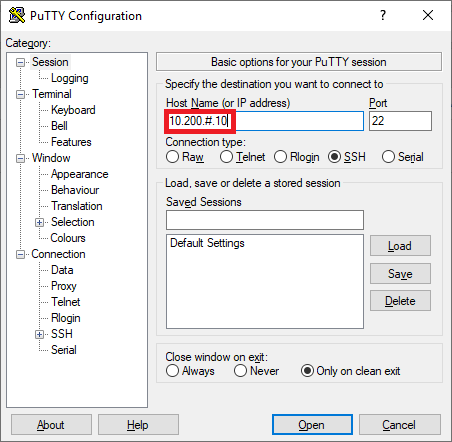
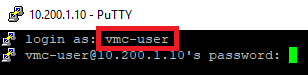
* 1. Access to your S3 bucket: **Use an existing IAM role**
  2. IAM role: **arn:aws:iam::229790159028:role/service-role/AWSStorageGatewayS3BucketAccess**

|  |
| --- |
| If you would like to review the IAM policy, you can find it here: <https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/AWSStorageGatewayS3BucketAccess>; however, the permissions are only viewable in JSON format. |

1. On the next page, click **Edit** next to **Allowed Clients**, and then enter the IP address of your Amazon Linux 2 Workload VM, which will serve as the NFS client for this exercise  
   
   1. Allowed clients: **10.200.#.10**, where ‘#’ is your student number
   2. Click the **Close** button on the right to finalize the allowed client configuration
   3. Review the rest of the settings, and then click **Create file share**
2. If all steps were completed successfully, you will see the following message:  
     
     
   and the share’s status should almost immediately update to **Available**  
   

### Mount your file share

We will now mount our Storage Gateway File Share via NFS v4.1 from our Workload VM, and then write some data to it.

1. In your virtual desktop, open PuTTY and enter the IP address of the Amazon Linux 2 Workload VM, and then click **Open**  
   
   1. Host Name (or IP address): **10.200.#.10**, where ‘#’ is your student number
2. Enter the credentials for your SSH session with the Workload VM:  
   
   1. login as: **vmc-user**
   2. password: **Amazon1!**
3. Mount the Storage Gateway file share via the following command:  
   **sudo mount -t nfs -o nolock,hard 10.200.#.11:/vmc-lab-resiliency-20191111220140298500000003/student### /mnt**  
   where ‘#’ in both the IP address in the middle and the student directory near the end is your student number

|  |
| --- |
| Example for student001:  sudo mount -t nfs -o nolock,hard 10.200.1.11:/vmc-lab-resiliency-20191111220140298500000003/student001 /mnt |

|  |
| --- |
| The mount operation should complete almost instantaneously. If the File Share’s Allowed Clients IP address was configured incorrectly in the previous section, the mount operation will hang and eventually timeout. |

1. Verify that the Storage Gateway File System mounted properly via the following command:  
   **mount | grep nfs4**  
     
   if successful, you will see output similar to this:

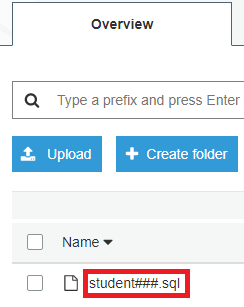
|  |
| --- |
| 10.200.#.11:/vmc-lab-resiliency-20191111220140298500000003/student### on /mnt type **nfs4** (rw,relatime,vers=4.1,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=10.200.#.10,local\_lock=none,addr=10.200.#.11) |

1. Generate a backup of all databases in the locally installed MariaDB database engine via the following command:  
   **mysqldump --all-databases > /mnt/student###.sql**, where ‘#’ is your student number  
     
   If successful, there will be no output.
2. Next, let’s verify that the backup file was created successfully via the following command:  
   **ls -l /mnt**  
     
   If successful, you will see output similar to this:

|  |
| --- |
| total 2  -rw-rw-r-- 1 vmc-user vmc-user 1396 Nov 20 05:56 student###.sql |

1. Let’s check out the backup file’s contents via the following command:  
   **cat /mnt/student###.sql**, where ‘#’ is your student number  
     
   If successful, you will see output similar to this:

|  |
| --- |
| -- MySQL dump 10.14 Distrib 5.5.64-MariaDB, for Linux (x86\_64)  --  -- Host: localhost Database:  -- ------------------------------------------------------  -- Server version 5.5.64-MariaDB  /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;  /\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;  /\*!40101 SET NAMES utf8 \*/;  /\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;  /\*!40103 SET TIME\_ZONE='+00:00' \*/;  /\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;  /\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;  /\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;  /\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;  --  -- Current Database: `test`  --  CREATE DATABASE /\*!32312 IF NOT EXISTS\*/ `test` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;  USE `test`;  /\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;  /\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;  /\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;  /\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;  /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;  /\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;  /\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;  -- Dump completed on 2019-11-20 5:56:24 |

1. Now let’s check out the backup file from the S3 bucket:
   1. URL: <https://s3.console.aws.amazon.com/s3/buckets/vmc-lab-resiliency-20191111220140298500000003/?region=us-west-2&tab=overview>
   2. Find and click on the **student###** directory, where ‘#’ is your student number
   3. You will see the **student###.sql** file, where ‘#’ is your student number  
      

|  |
| --- |
| You cannot open the file from the S3 bucket because public access has intentionally been blocked. |

## Conclusion

In this module, you have:

* Configured and activated an AWS Storage Gateway that sends its traffic across the VMware Cloud ENI to VPC Endpoints instead of over the public internet
* Configured an NFS file share backed by a highly available and highly durable Amazon S3 bucket
* Uploaded a new database backup file

In addition to File Gateway mode, AWS Storage Gateway also offers a Tape Gateway mode that provides a virtual tape library (VTL) interface to Amazon Glacier and Amazon Glacier Deep Archive, which could be used to further enhance the resiliency of your business’ critical data. [Many of our backup & recovery partners support Tape Gateway as a VTL target](https://docs.aws.amazon.com/storagegateway/latest/userguide/Requirements.html#requirements-backup-sw-for-vtl), including: Arcserve Backup, Bacula Enterprise, Commvault, Dell EMC NetWorker, IBM Spectrum Protect, Micro Focus (HPE) Data Protector, Microsoft System Center, Microsoft Data Protection Manager, NovaStor DataCenter/Network, Quest NetVault, Veeam Backup & Replication, Veritas Backup Exec, and Veritas NetBackup.

# Module - Analytics

In this module, you will get to learn how to take advantage of AWS data analytics services to gain insights from data. With VMware Cloud on AWS service, you can seamless migrate on-premises databases to the cloud. With data gravity removed from your on-premises datacenter, you can gain insights from your migrated workload through usage of native AWS analytics services for data warehousing and data lake. Please follow through the customer scenario below and work through the solution architecture with integrated AWS services.

## Customer problem

Your CIO is implementing a “cloud first” strategy and demands for all VMware virtualized applications to migrate to the cloud. With datacenter evacuation being the first step of the journey to the cloud – you have successfully migrated the Postgres database to VMware Cloud on AWS. As a proactive infrastructure and cloud architect, you are taking a fresh look at data analytics to empower your business intelligence team to consistently gain insights faster, simpler and cheaper. You have then been given the following requirements/guidance from your CIO on the proposal:

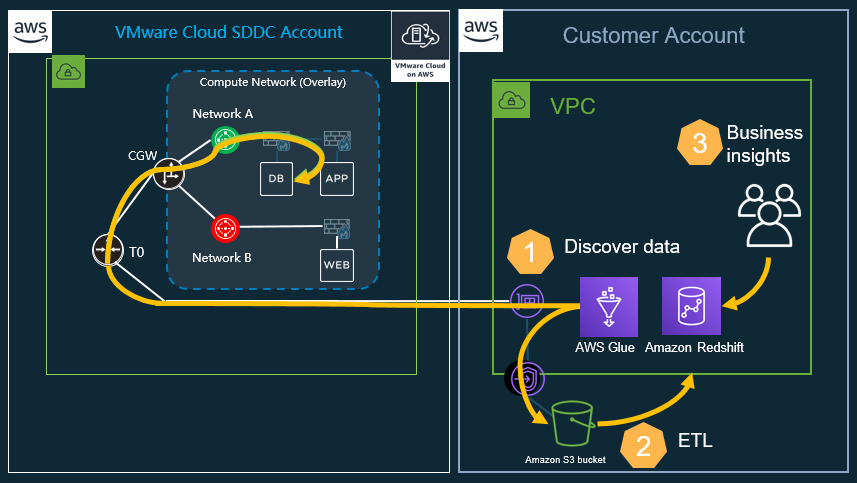
* Solution needs to have solid data pipeline orchestration
  + decouple analytics storage from transactional database
  + decouple compute from storage
  + orchestrated extract, transform & load
  + go serverless whenever possible
* Solution needs to be highly available, highly performant, highly resilient & cost effective
* Solution needs to be future proof, with flexibility of services & toolchains for integration & data exploration

## Proposed solution

After some extensive research, you have come to learn that the following AWS services can be leveraged to address all the requirements:

* AWS Glue: a fully managed extra, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics.
* Amazon S3: Amazon S3 provides an optimal foundation for a data lake because of its virtually unlimited scalability. It is designed to provide 11 9’s of durability. It has scalable performance, ease-of-use features, and native encryption and access control capabilities.
* Amazon Redshift: Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools.

The solutions architecture below illustrates how AWS Glue, Amazon S3 and Amazon Redshift can be integrated seamlessly to work with migrated database VMs running on VMware Cloud on AWS SDDC cluster:



1. AWS Glue supports various data sources including JDBC connection to database sources running on-premises, RDS or VMs in VMware Cloud on AWS SDDC cluster. With the migrated Oracle database VM running in VMware Cloud on AWS SDDC, Glue can connect to the database through the JDBC connection and discover available table(s) and their respective schemas.
2. ETL (extract, transform & load) jobs can be scheduled through Glue. With the database table identified and schema discovered through the Glue crawler, data can be copied from source database to Amazon S3. Format conversion can be done easily by selecting the desired format such as Parquet. Lastly, extracted & transformed data with columnar optimized Parquet format can be analyzed through Athena service (ad-hoc SQL-based queries). Regular reporting can also be done through Redshift Spectrum.
3. Lastly, business intelligence teams can access query data from Redshift data warehousing cluster to gain business insights.

You are now ready to configure your analytics workflow by performing the following tasks:

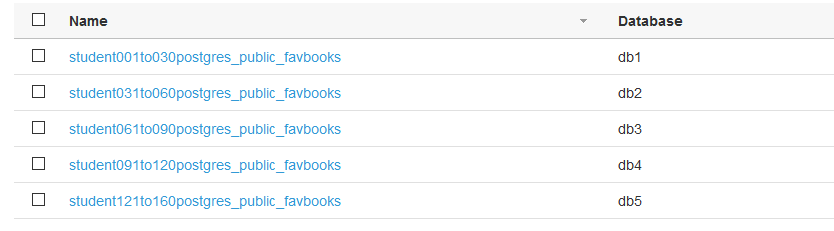
1. Decouple production transactional database data storage for analytics: create ETL (Extract Transform and Load) jobs by copying data from Postgres database running on VMware cloud on AWS into Amazon S3 bucket, and converting the data format to Parquet
2. Add copied data in Amazon S3 bucket to the AWS Glue catalog by creating and running a crawler against the S3 bucket folder where the data was copied
3. Ingest data from S3 to Redshift data warehouse cluster
4. Query data against Redshift table

AWS Glue environmental overview

There are five databases created within AWS Glue, with the following access instructions:

|  |  |
| --- | --- |
| **Database** | **Students** |
| db1 | 001-030 |
| db2 | 031-060 |
| db3 | 061-090 |
| db4 | 091-120 |
| db5 | 121-160 |

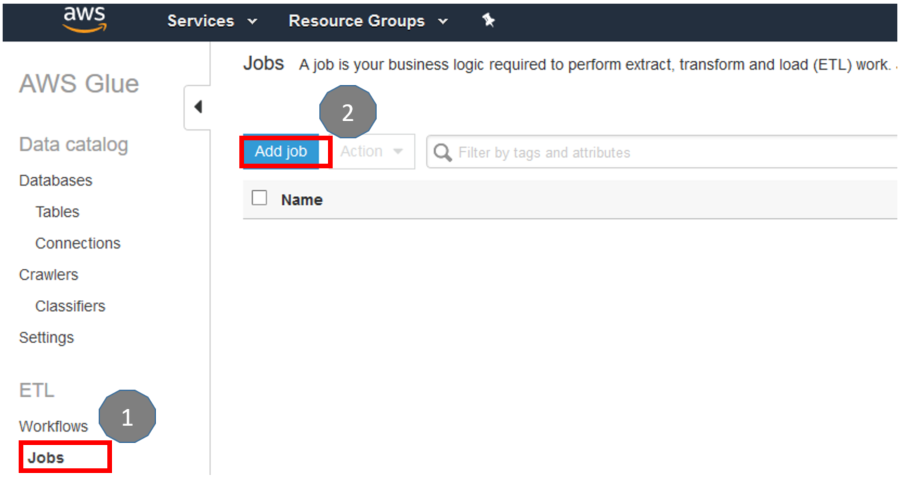
The tables containing favorite books voted by general public have been discovered by Glue crawlers. The metadata for the tables have been discovered and populated in the glue database. During the lab exercise, please be mind to choose only the table that you are assigned to, based on your student ID. If you do not know your student number, please ask your lab captain. For example, if you are student035, you will work with **“student031to60postgres\_public\_favbooks”** table in database **“db2”**:



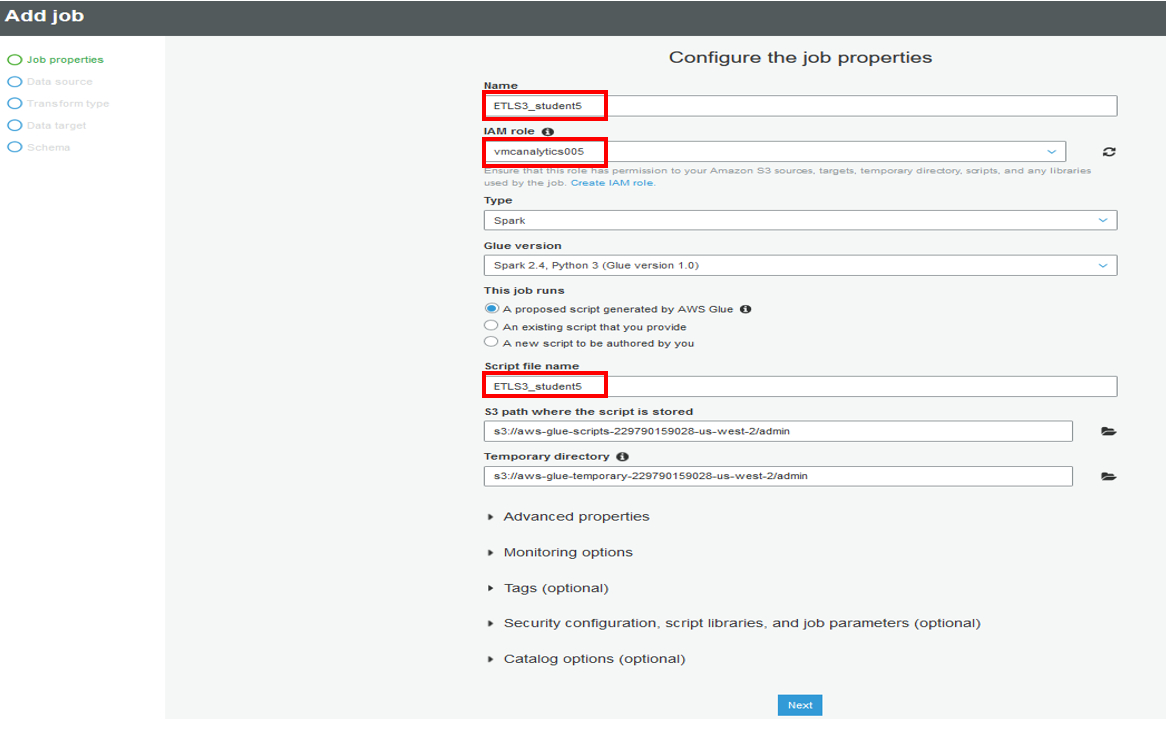
**Step 1) Decouple storage**

In the AWS Management Console, find “Glue” service and click into the Glue service user interface

Click on “Jobs” under Workflows and then click on “Add job” to create an ETL job to copy data from the Postgres database running on VMware Cloud on AWS SDDC cluster



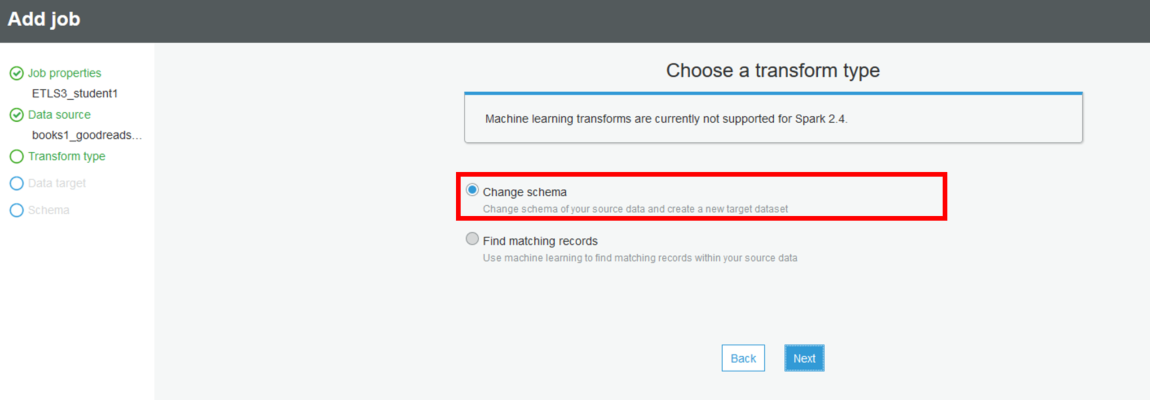
For the job properties, ensure the “Name”, “IAM role” and the script name match the example format below – make sure you use your student number in the name. The example below shows the naming standard for “student1”. If you are student 5, then your “Name” entry will be ETLS3\_student5, and the IAM role you will select will be “vmcanalytics005”. And the script name should read “ETLS3\_student5”:



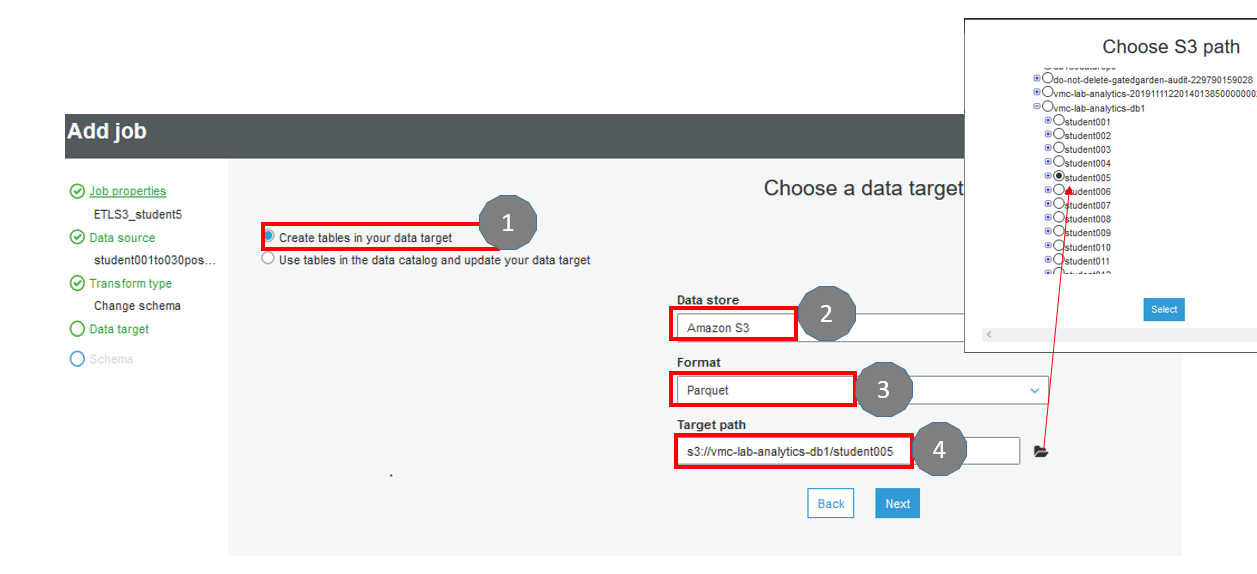
For the data source, select the table name corresponding to your student number. The example below shows “books1\_goodreads\_books1” for student 1:



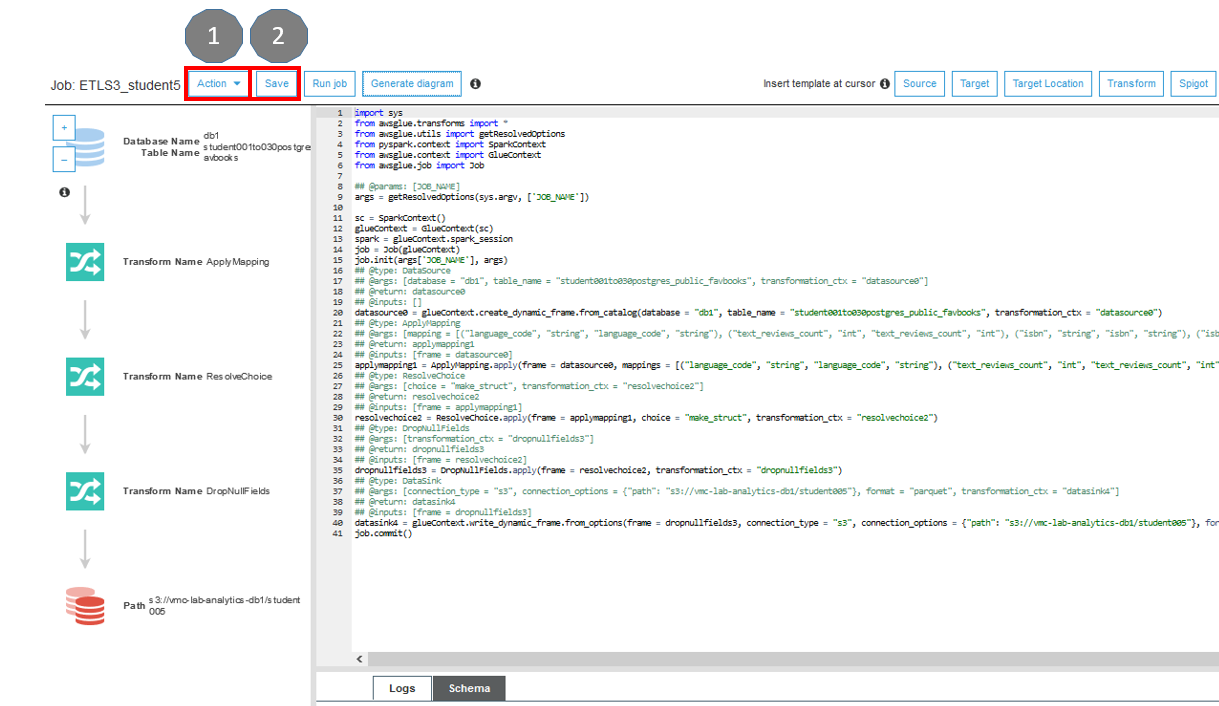
Select “Change schema” for the transformation type



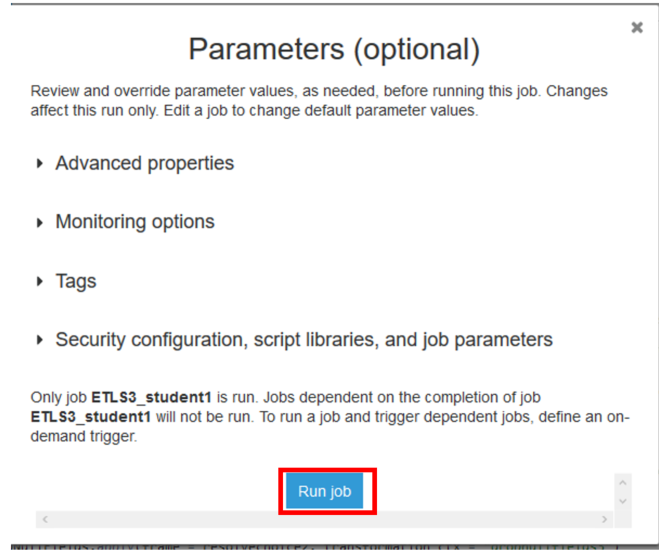
For data target, we will create new tables in the target by selecting “Create tables in your data target”. We are going to use Amazon S3 as our data lake analytics storage. Select “Amazon S3” as Data store type, “Parquet” as the format we want to transform our source data to, and select your designated folder within the pre-created S3 bucket for storing the transformed data – as an example below, for student 5, the bucket to select is “vmc-lab-analytics-db1”, folder “student005”:



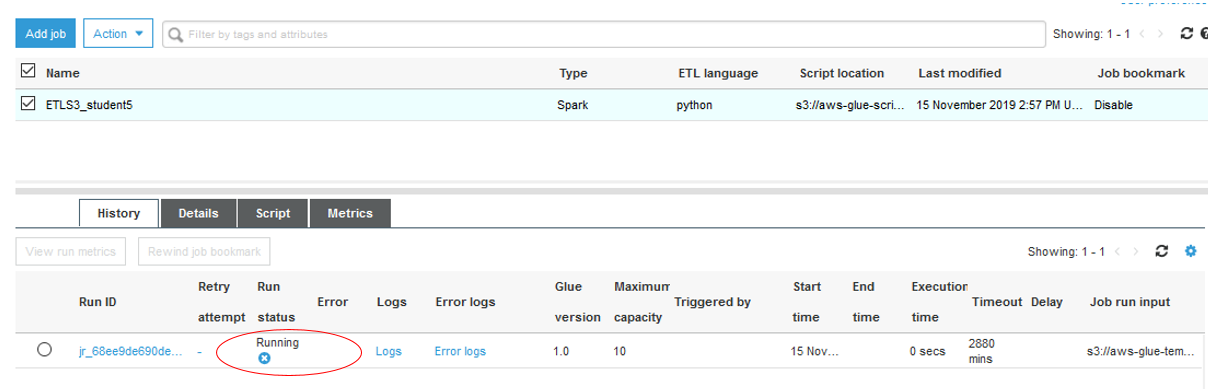
Glue provides the ability for you to remove unwanted columns in the database table, or re-order specific columns of choice. For this step, you are free to modify how the table will look in the target – when you have made your modifications by removing or re-ordering columns, click on “Save job and edit script” to proceed to the next step:

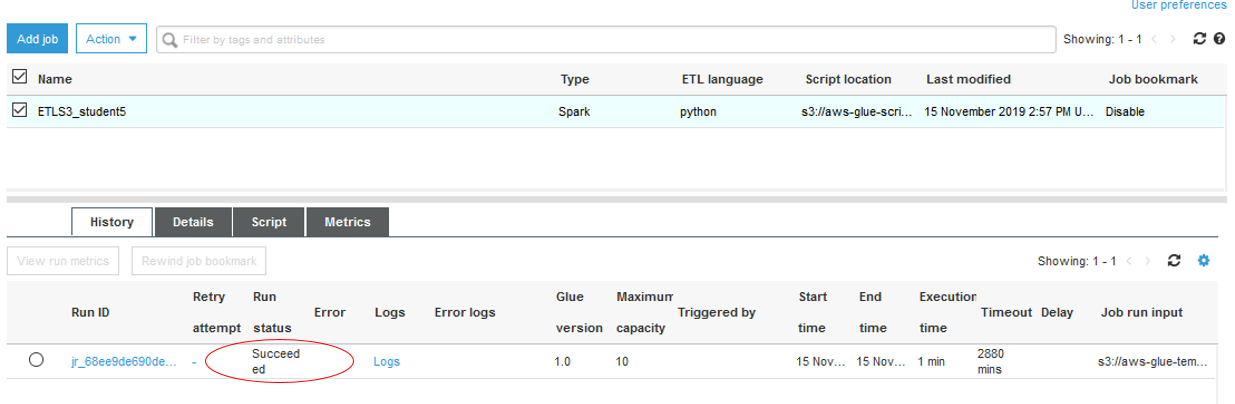
In this step, you can click on each of the icons created for your job to see the Spark script that is generated for you. We are not modifying the script so click on “Save” to save the job, follow by “Run job”. 

Click on “Run job” and then close the script editor window by clicking on “x” on the upper right hand corner:

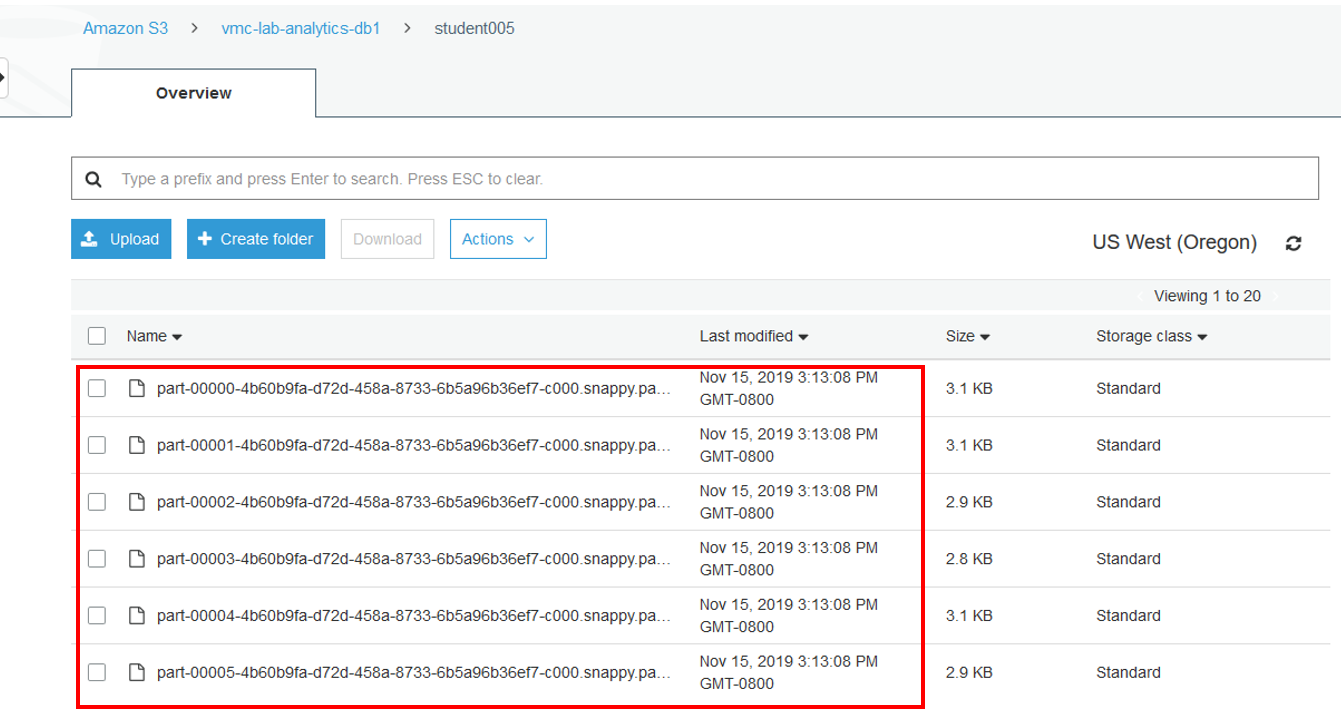


You can now monitor the status of the running job and watch for it to finish:



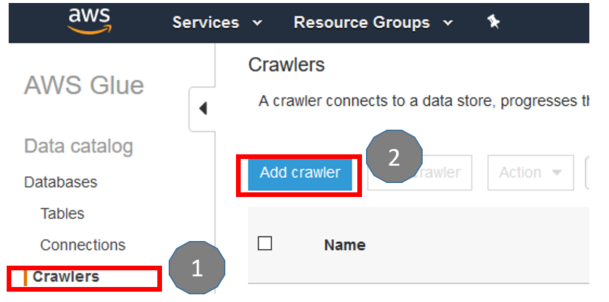


If you job completes with the status of “Succeeded”, you are now ready to check the written Parquet transformed data in your respective student folder in the S3 bucket:

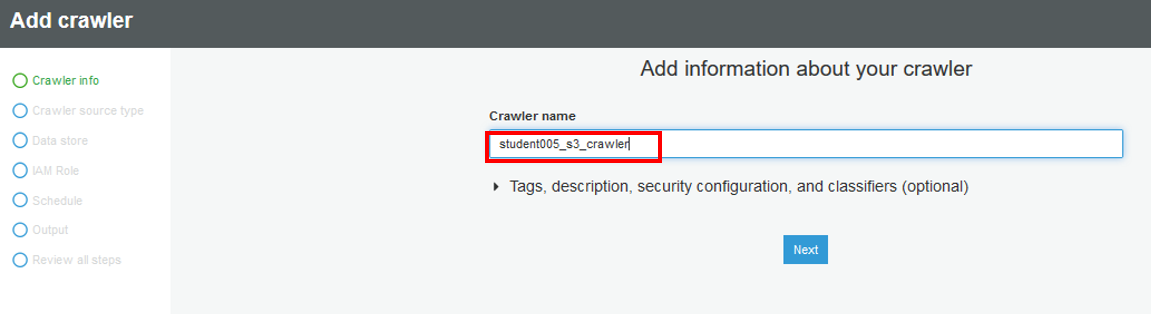


Step 2) Update Glue Catalog by crawling the newly copied data in Amazon S3 bucket

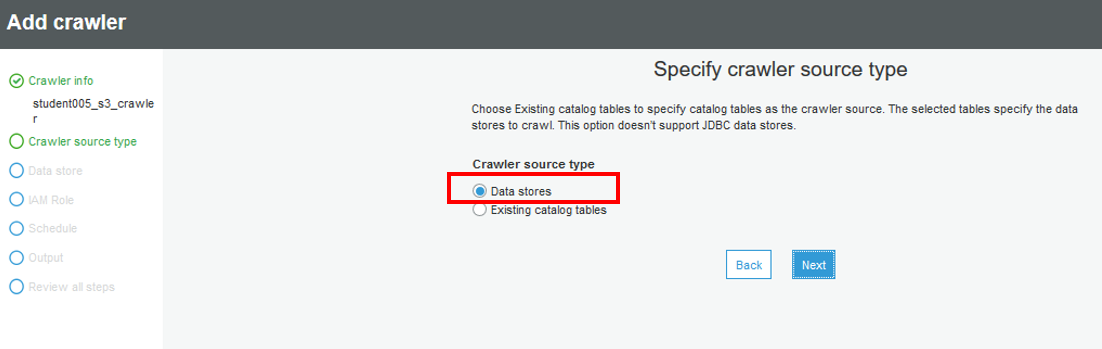
From the AWS Management Console, go into the Glue service user interface, and then click on “Crawlers” and click on “Add crawlers”



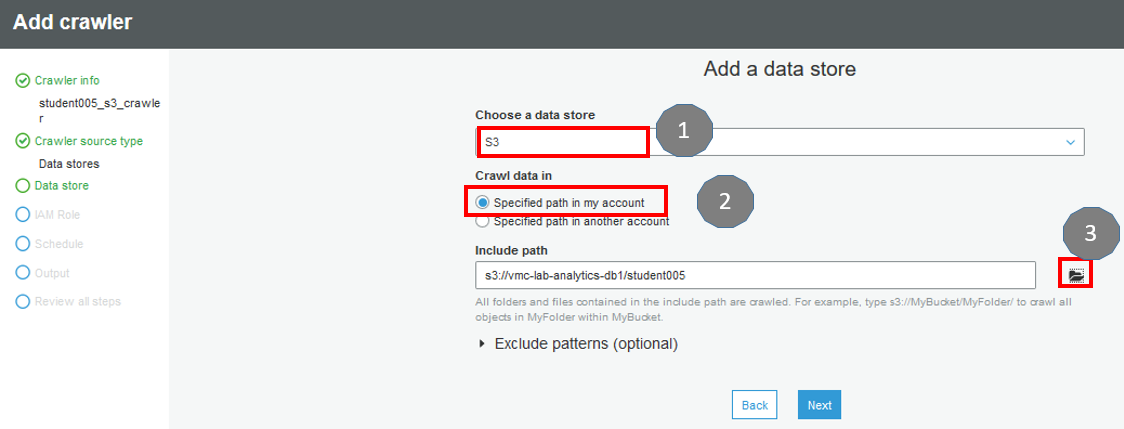
Name your crawler based on your student number – example for student is the “student1\_S3\_crawler”:

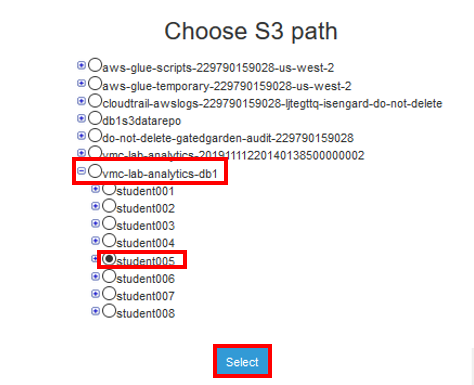


Select “Data stores” as the source type



Since our extracted and transformed data in Parquet format is stored on Amazon S3, choose S3 as the data store. The S3 bucket is owned by the AWS account you are currently using for this module, therefore, we will have the crawler crawl the data in the current AWS account. The last step here is to select the appropriate S3 bucket folder based on your student ID.

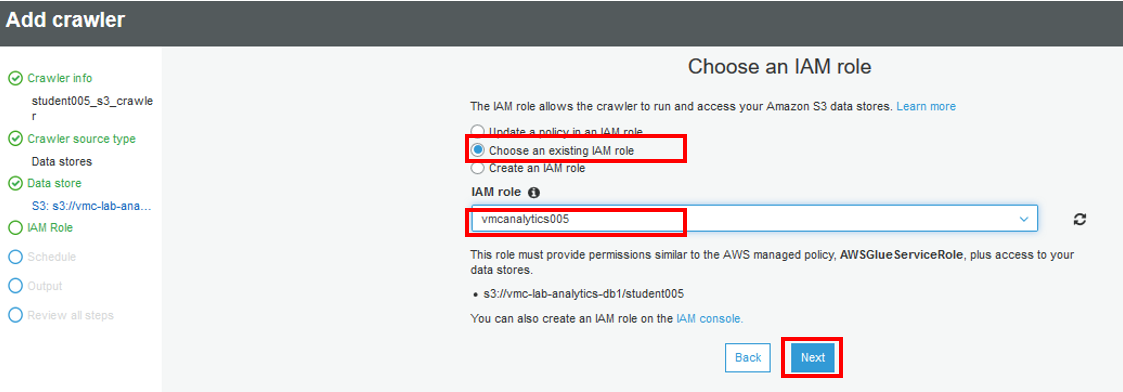




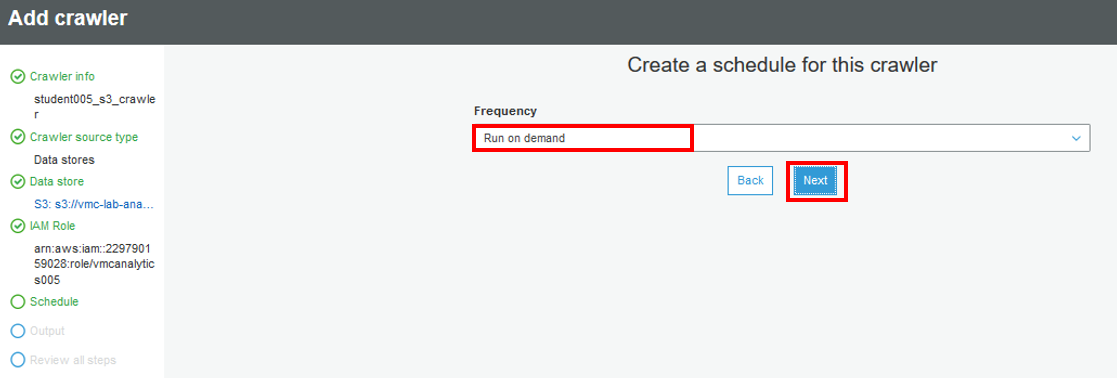
For the next step, we won’t be adding another data store so select “No”



Select the appropriate IAM role based on your student ID, for example, student005 will choose vmcanalytics005 IAM role:



Set up the crawler to “Run on demand”. In real life scenarios, you would want to obtain business requirements and configure the crawler to run based on schedule or specific triggers such as new data uploaded to S3 bucket.



Next we will choose which Glue database to populate the catalog with the newly stored data on S3 discovered by the crawler. Please choose your database based on the DB group your student ID belongs to – for example, student005 will choose “db1” as it belongs to Group 1, and enter in the prefix “student005s3” to uniquely label the discovered data table in the Glue catalog:

**Group1 (“db1”):**

* + students001-030

**Group2 (“db2”):**

* + student031-090

**Group3 (“db3”):**

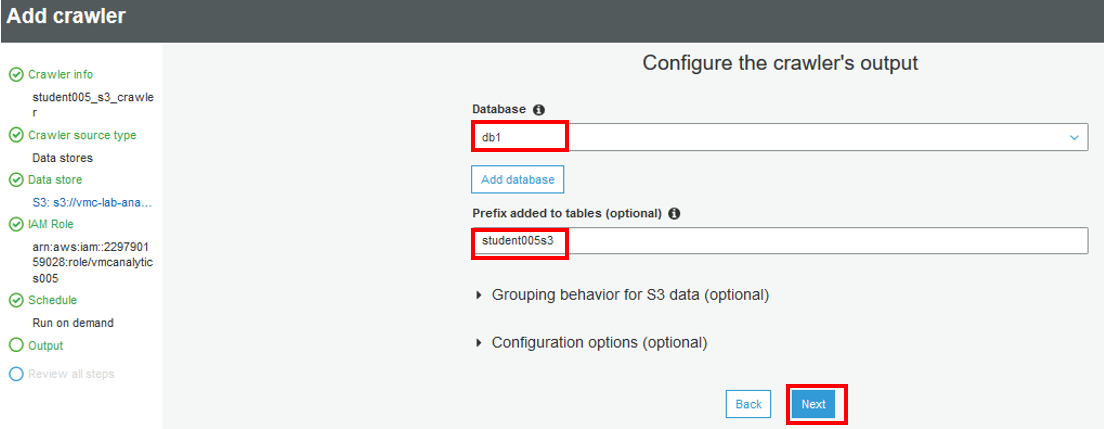
* + student061-090

**Group2 (“db4”):**

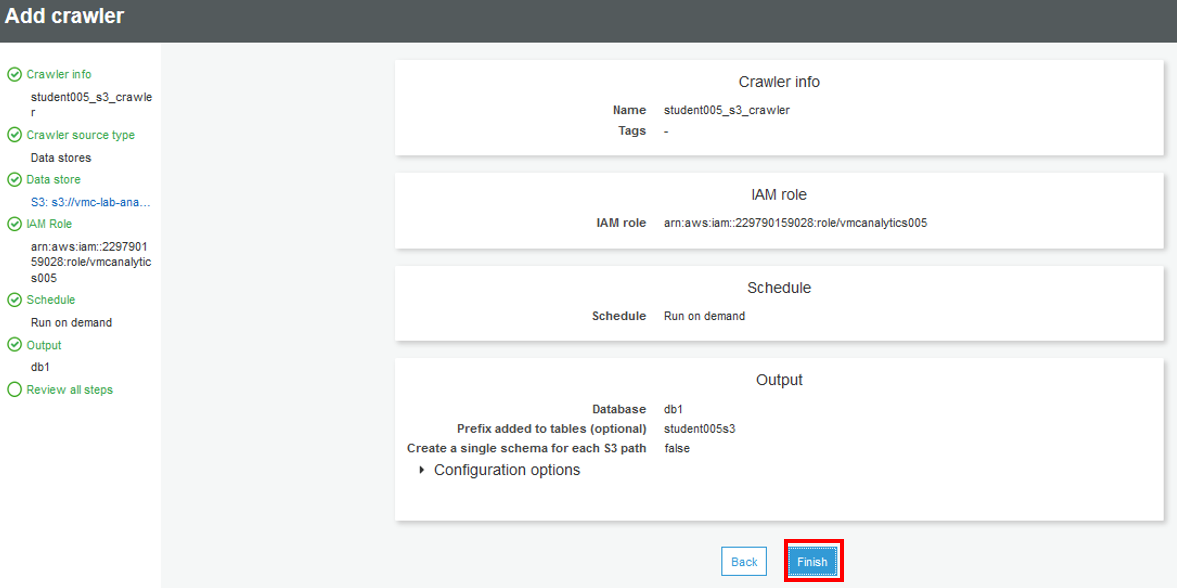
* + student091-120

**Group2 (“db5”):**

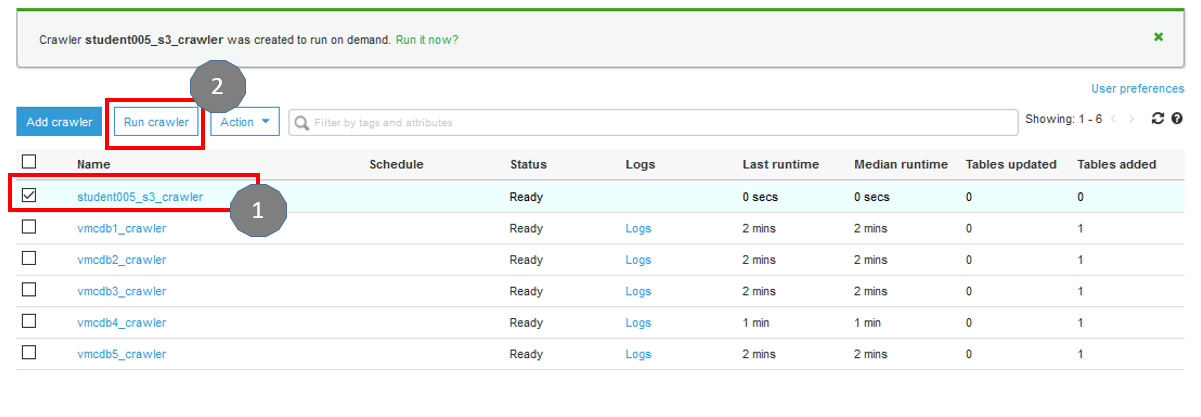
* + student121-160



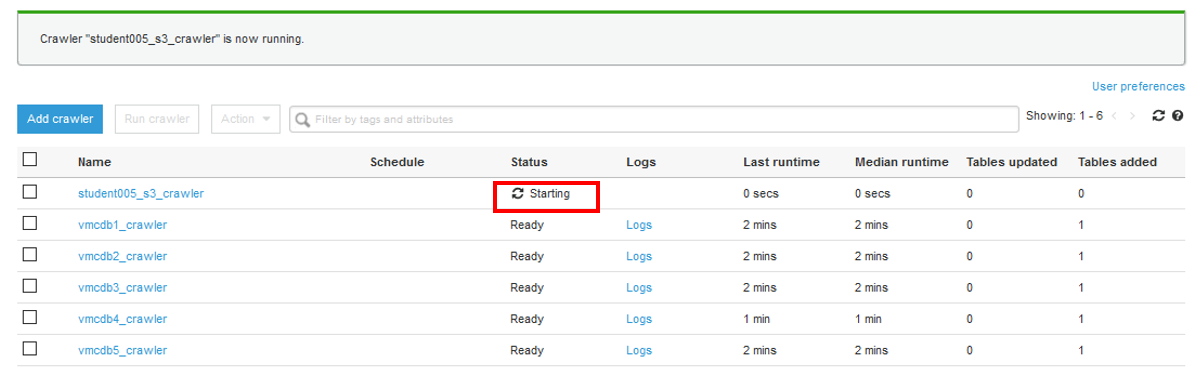
Review all settings and click “Finish” to complete the creation of S3 crawler

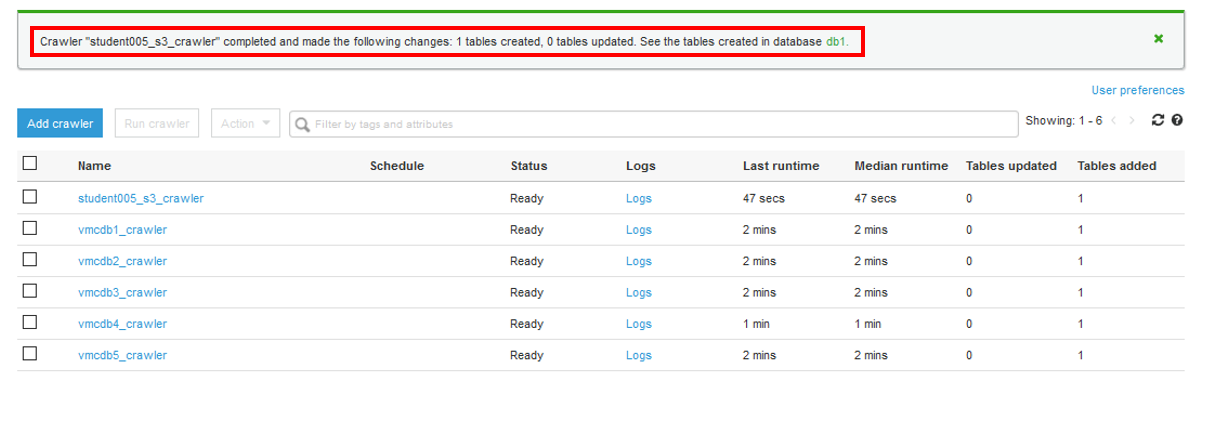


You are now ready to run the crawler to discover newly stored data on your S3 bucket and get Glue catalog updated!

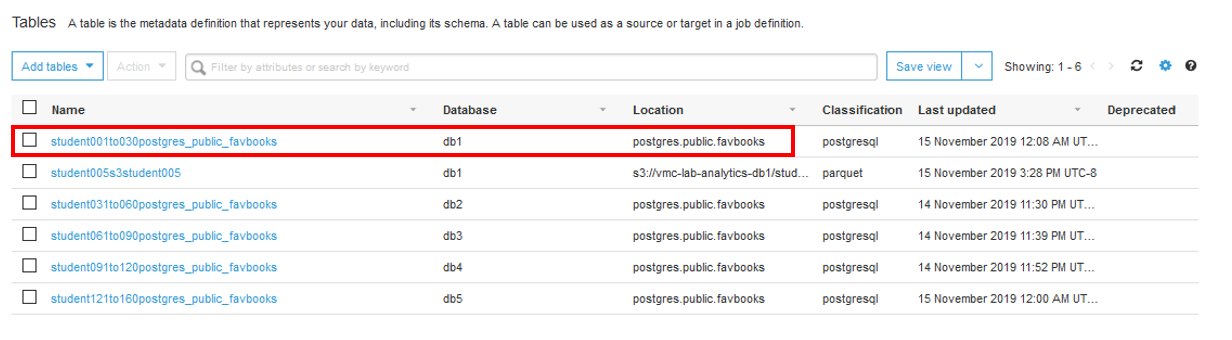


Monitor the job status until it shows “Stopping”. Once the status for the job changes from “Stopping” to “Ready”. You should see the Glue database catalog getting updated with the newly discovered data source on S3:



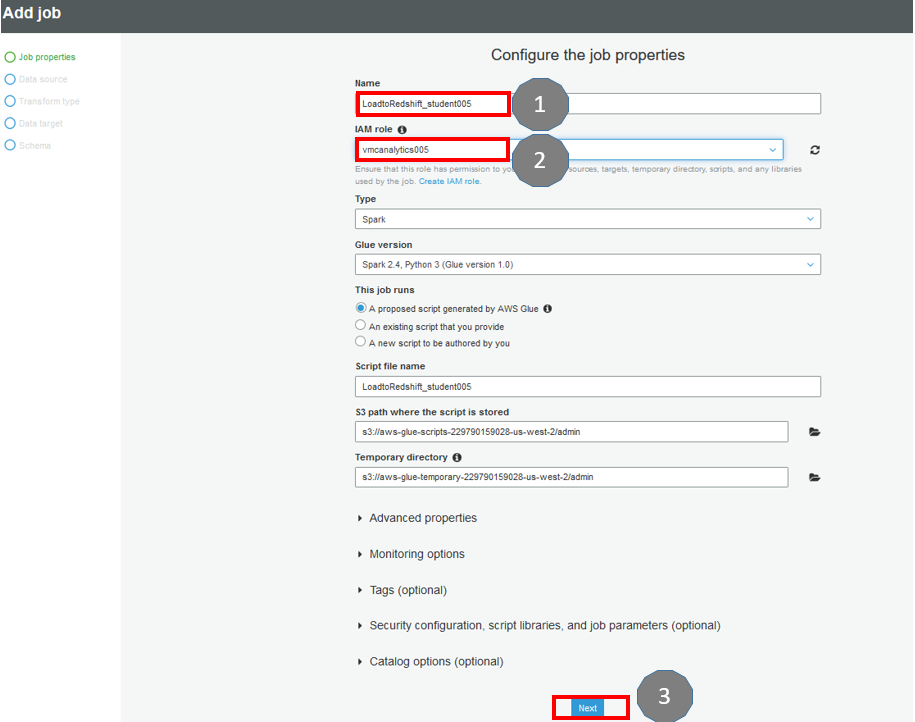


Check the Glue database catalog under “Tables” and confirm that a new catalog entry has been added for your respective database, and has the location of S3:

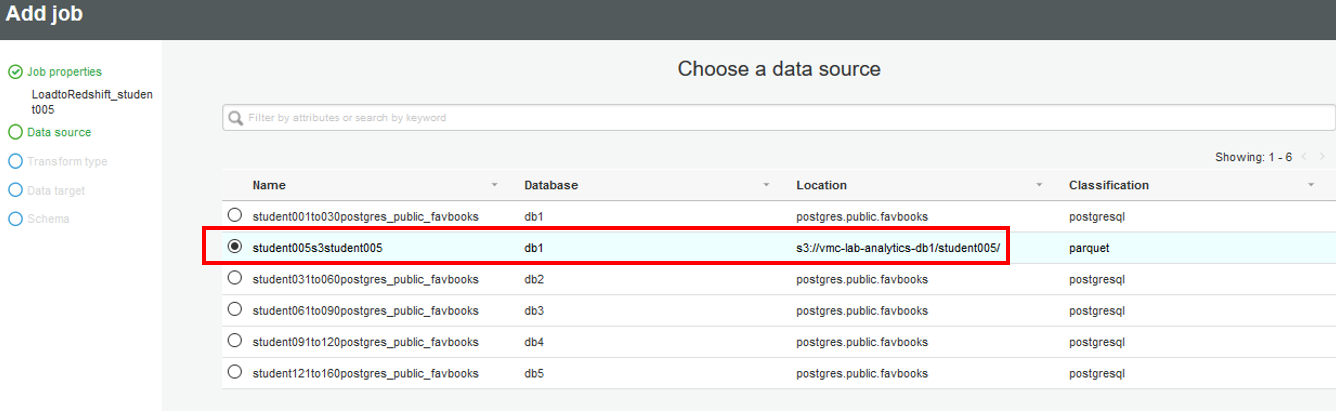


Step 3) Ingest data from S3 to Redshift

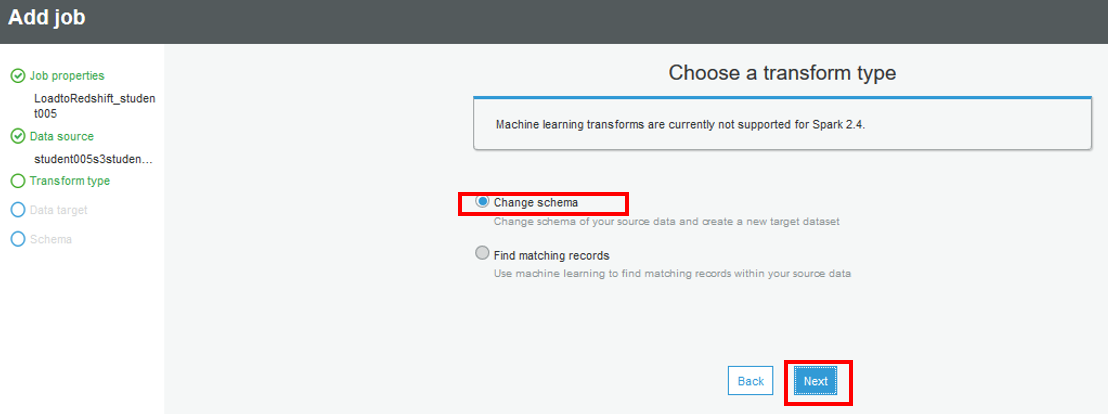
Create a new job to copy data from S3 bucket to Redshift data warehouse cluster. Ensure the naming convention is followed for the job name based on your student ID. For example, Student 1 will have the job name of “LoadtoRedshift\_student1”, and the role of “Module4GlueRole1” will be selected:



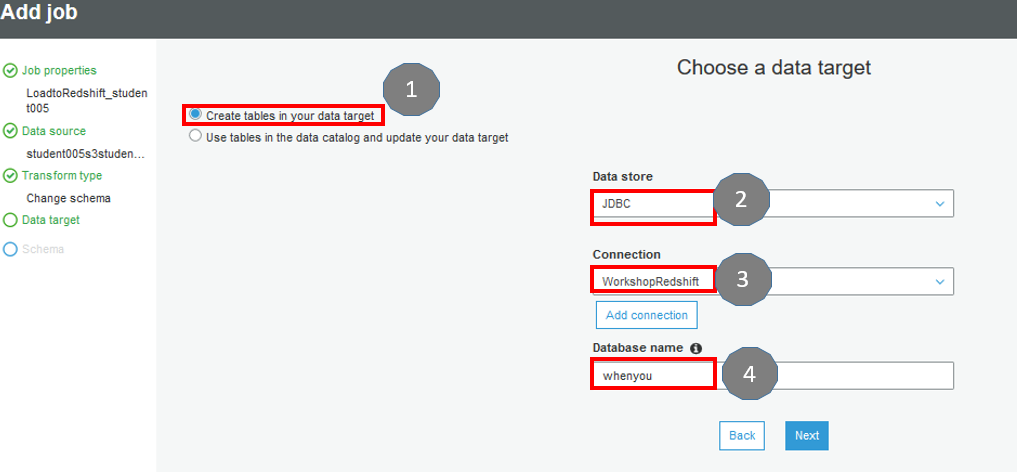
For data source, ensure your corresponding S3 based table from the catalog is selected – Student 1 will select “1student” from “db1”



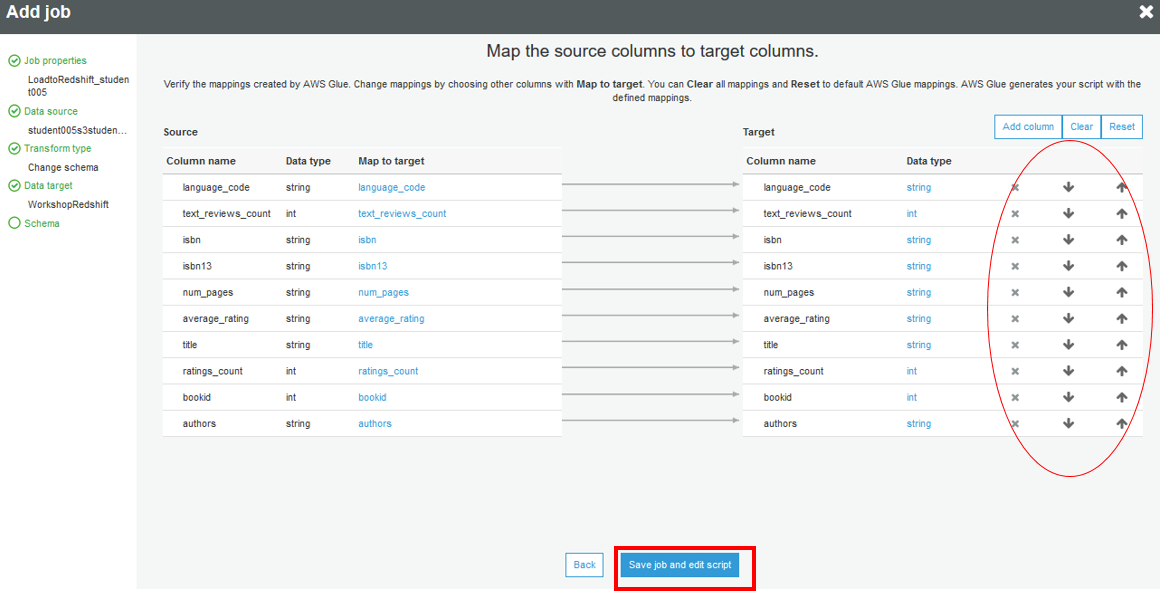
Ensure “Change schema” is selected then click “Next”



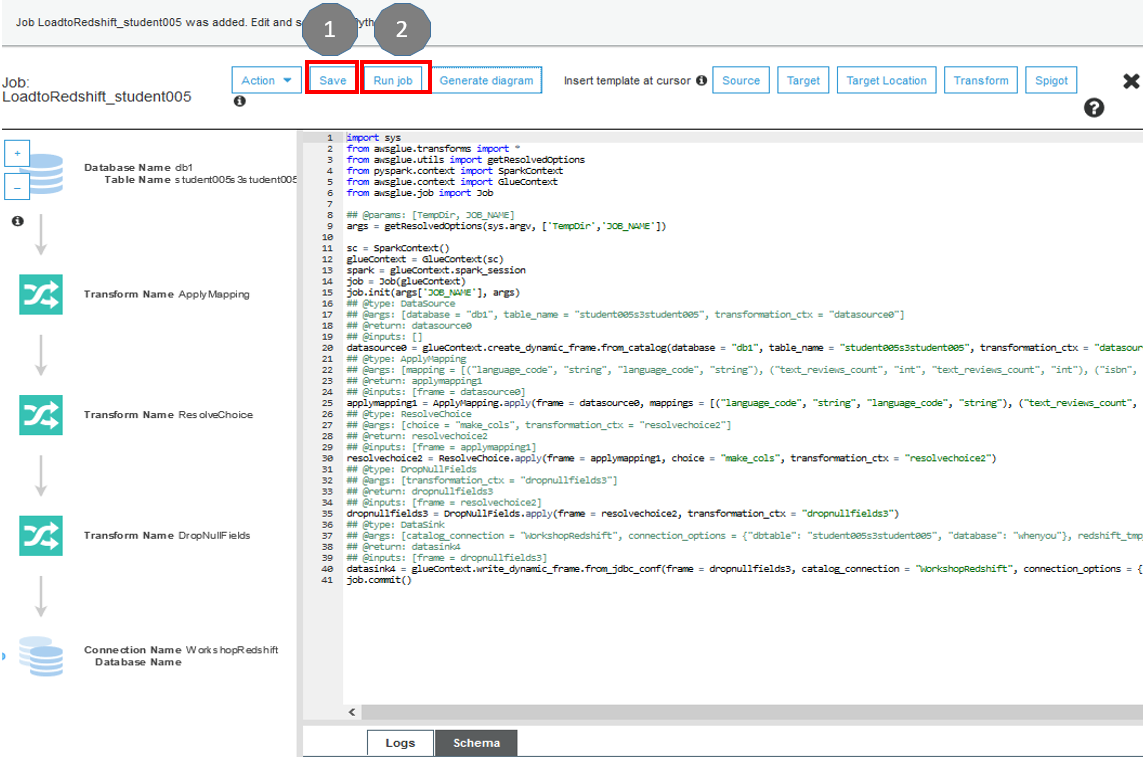
We want to create new tables in the Redshift data warehousing cluster – ensure “Create tables in your data target” is selected, and select the data store parameters as shown in diagram below:

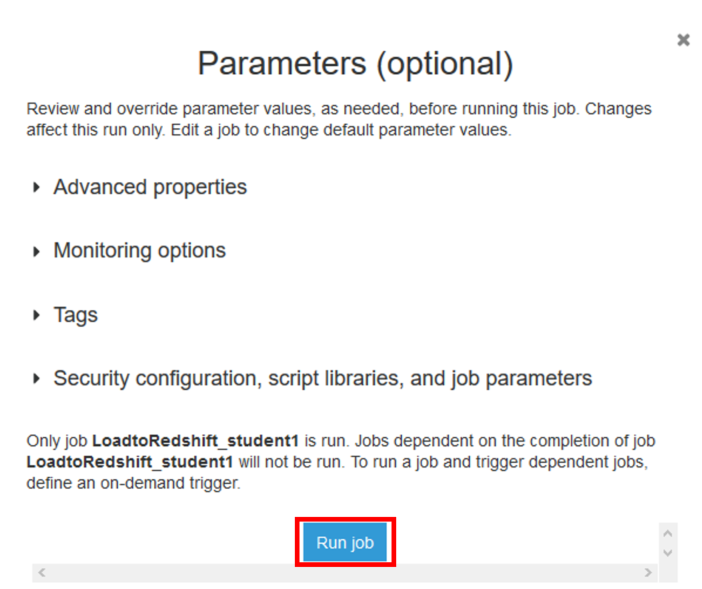


You have the flexibility to remove specific column(s) and/or re-arrange the ordering – when finished, click “Save job and edit script”

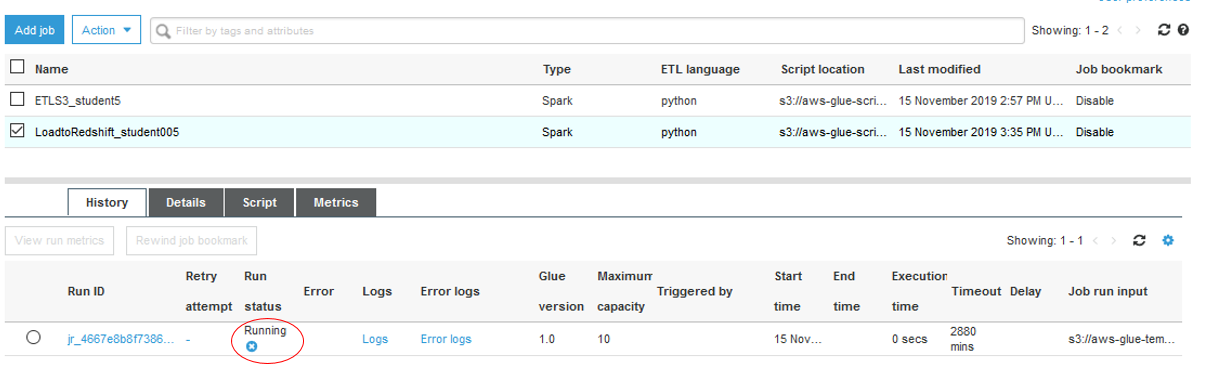


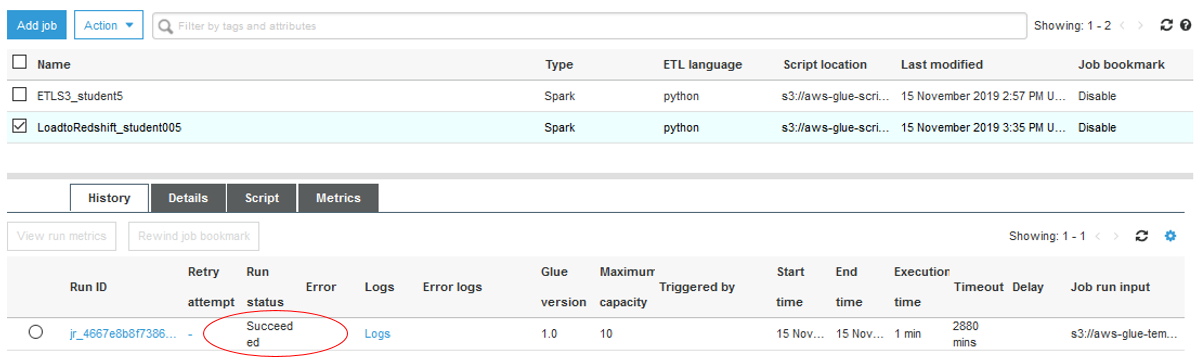
For the script generated, familiarize yourself with the various steps in the job. We are not modifying the script for this exercise. If you choose to modify it, you are responsible for troubleshooting any failed job run ☺. When ready, click “Save” follow by “Run job”





Check to ensure the job runs to completion with “Successful” status at the end





You have successfully created your data analytics catalog, and data ETL pipeline to decouple transactional database with data warehousing.

# Thank You

Thank you for participating in the workshop. If you have any questions about VMware Cloud on AWS please reach out to the session support team personnel. You can also reach us at [aws-vmware-cloud@amazon.com](mailto:aws-vmware-cloud@amazon.com)

<https://aws.amazon.com/vmware>

END OF WORKSHOP LAB GUIDE

1. This the VPC that is connected to your VMware Cloud on AWS SDDC VPC. [↑](#footnote-ref-1)