

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

# Clouducate

## Enabling Cloud-based IT Hands-on Education

Norbert Monfort & Robert Fortunato

# Clouducate Overview

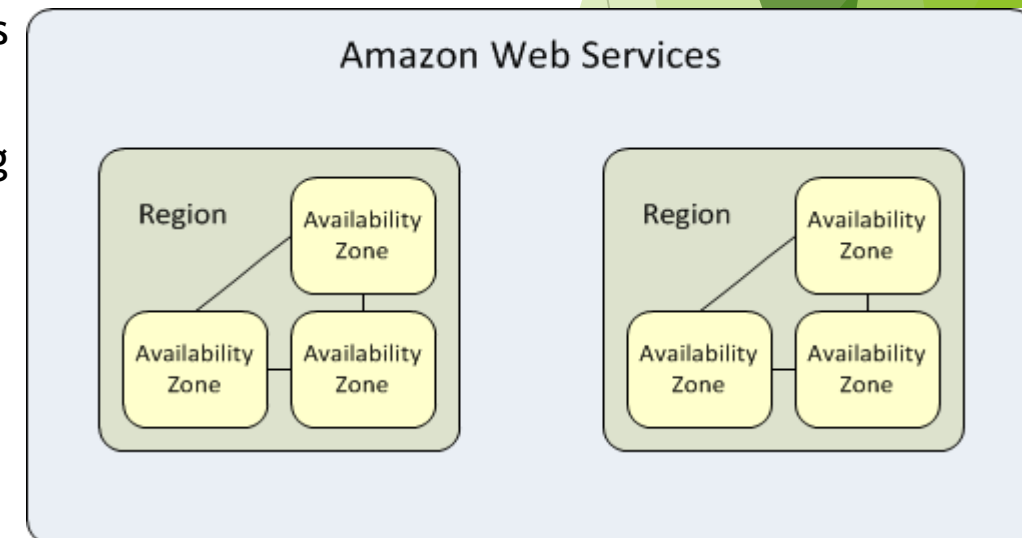
AWSVPCB (AWS Virtual Private Cloud Builder) is a set of BASH Shell scripts designed to create a small enterprise IT-like environment in AWS to provide college-level students with hands-on experience individually working on IT problems. These scripts are now focused on the use of the AWS Academy Learner Lab but still support running outside of the AWS Academy Learner, with configuration adjustments.

These scripts work in conjunction with AWS images (AMIs) with pre-loaded applications and a method for automatically changing the environment for assignment purposes. These scripts are open source and free to be used for any purpose. These scripts, the associated AMIs and applications they house make up what is called the "Clouducate Suite", which we continue to work on.



# AWS Concepts & Glossary

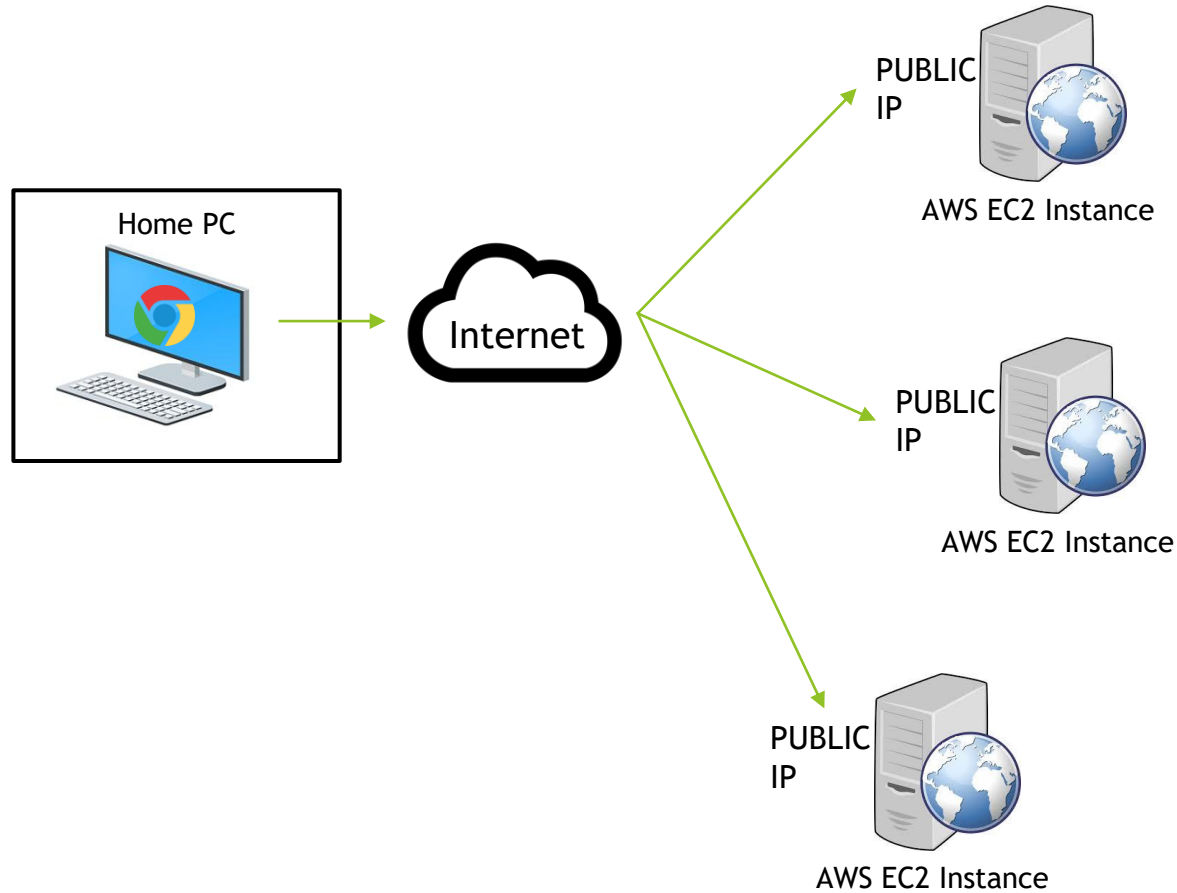
- ▶ AWS Academy - Courses and “Learner Lab” Offering for Universities (FIU has agreement)
  - ❑ The “Learner Lab” offers \$50 per student, and this is the preferred approach for using AWSVPCB
- ▶ AWS Region - One or more AWS data centers within a nearby area (within 250 miles) - By default, AWSVPCB uses “us-east-1” as all of the AMIs are in this region
- ▶ AWS Availability zone - One or more data centers within a region - AWSVPCB uses 2 AZs by default (“us-east-1A” - primary defined in vpcb-config and “us-east-1b” secondary defined in assignment json files)
- ▶ AWS VPC (Virtual Private Cloud) - Private network within AWS includes large private address space; can span Availability zones, but not regions
  - ❑ Used in AWSVPCB to simulate an Enterprise environment
- ▶ AWS VPC Subnet - Portion of VPC addresses used to group instances & services; Cannot span availability zones
- ▶ IAM User - It’s recommended to create an Admin level user, if using AWSVPCB outside of AWS Academy to avoid using ROOT user
- ▶ EC2 (Elastic Compute Cloud) Instance - A VM server in AWS (can be associated to a VPC subnet and a security group)
- ▶ S3 Storage - AWS’ Simple Storage Service (object storage via Internet access)
  - ❑ Used for Load Balancer logs and Professor DIAGLOG uploads



# AWS Concepts & Glossary

- ▶ ec2-user - Userid to login to your Linux EC2 instances
- ▶ Administrator - Userid to login to your Windows instances
- ▶ EC2 instance states - running, stopped and terminated (deleted)
- ▶ Routing table - Associated to VPC subnets - how to route network requests
- ▶ Security Group - Associated to EC2 instances/AWS services - firewall rules for access in & out of instance/service
- ▶ NAT Instance - A pre-configured EC2 instance that allows outbound access to Internet
- ▶ BASTION Instance - A pre-configured EC2 instance that allows RDP access into VPC
- ▶ Internet Gateway - Allows Internet access out of VPC
- ▶ Client VPN Endpoint - Legacy feature of AWSVPCB that allows VPN access into VPC - Associated to a VPC subnet
- ▶ Route 53 - AWS' DNS service - By default setup up with private zone (e.g. AWSVPCB.edu) for DNS resolution within VPC
- ▶ ELB (Elastic Load Balancer) - AWS' load balancer service (there are multiple options - we will utilize the Classic LB) - Associated to subnet & security group

# AWS Basic Config

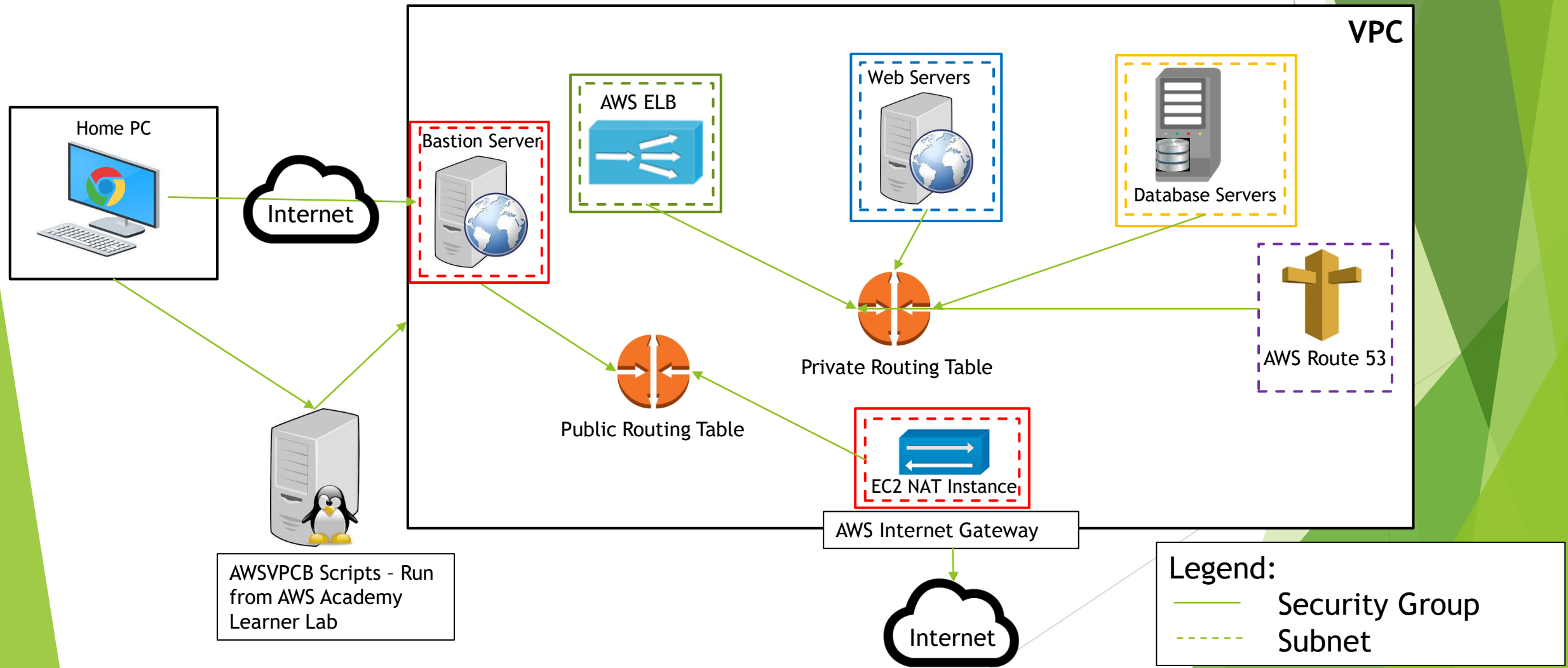


## ► Problems with this config:

- ❑ Not representative of Enterprise IT shops
  - ❑ All servers are exposed to the Internet (no private addressing)
  - ❑ No local DNS domain or servers
  - ❑ No firewall zones
  - ❑ No load balancing
  - ❑ No subnetting
- ❑ Overly simplistic - can't simulate real world problems

# AWSVPCB DEFAULT Config (vpc0)

AWSVPCB comes with 2 pre-defined VPC configurations (vpc0 & vpc1). The difference is how students connect into the VPC with vpc0 (default and recommended) shown below using a BASTION host and vpc1 displayed in the appendix using a client VPN.



# AWSVPCB Default Assignment Diagrams

AWSVPCB comes with 17 pre-defined assignments. Assignments 1-3 have no issues; these are the baseline or example assignments. Assignments 4-17 have one or more issues.

Below are the number of issues for each assignment:

Assignments 1-3: No issues

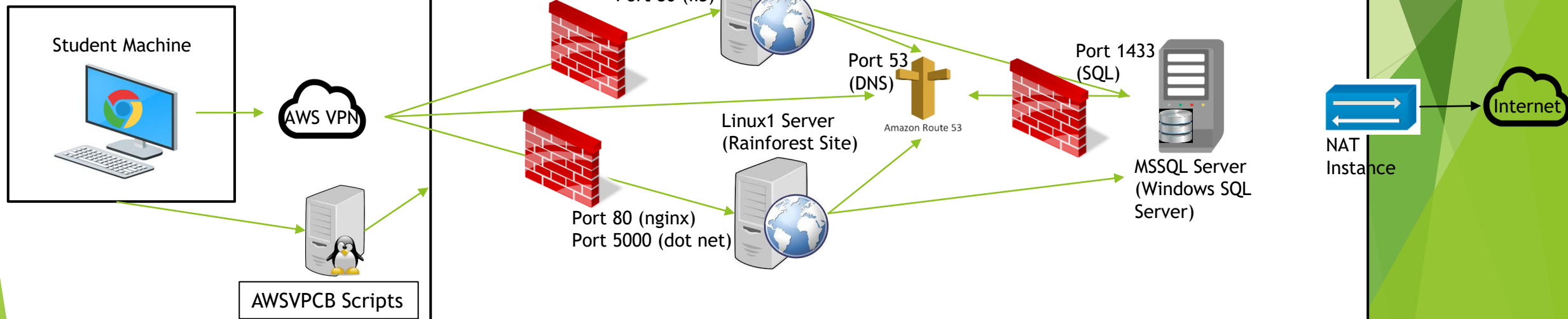
Assignments 12-17: 1 problem to fix

Assignments 4-8: 2 problems to fix

Assignments 9-10: 3 problems to fix

Assignment 11: 4 problems to fix (one is for extra credit)

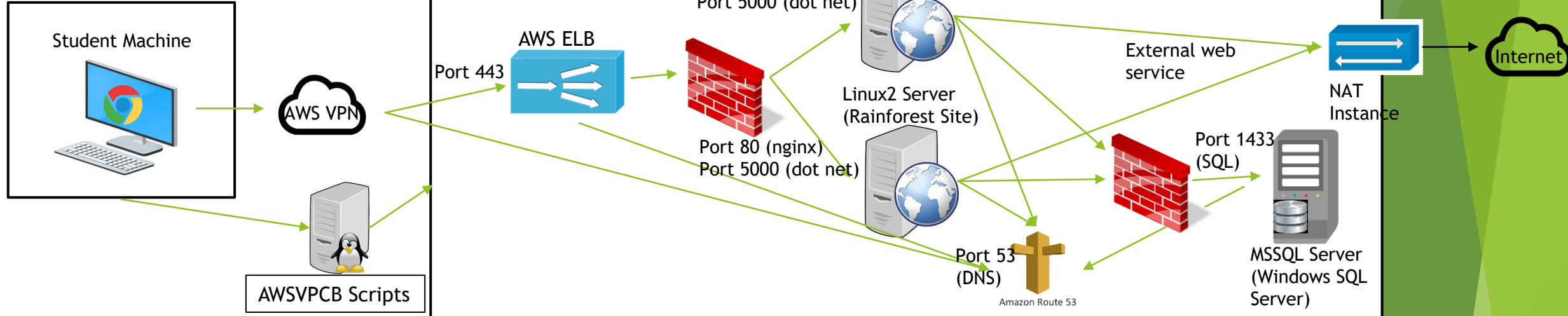
Assignments #1, #4, #5,  
#6, #7, #13, #14, #15, #17



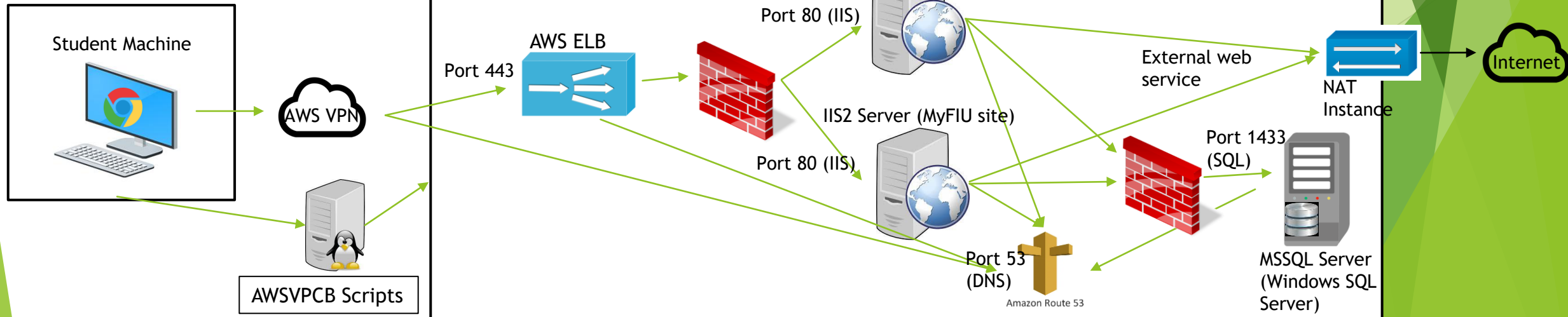


# AWSVPCB Default Assignment Diagrams

## Assignments #9, #11, #16 Only Rainforest App



## Assignments #8, #10, #12 Only MyFIU App





# AWSVPCB SCRIPTS

- ▶ In this course you will be provided with several scripts to manage your VPC
- ▶ 3 one-time scripts as follows, 1 recovery script and 1 diagnostics:
  - ❑ AWSVPCB.CONFIGURE - Only needed at start of course, but can be run again without issue; Student may be asked to select their professor & course, if vpcb-config is defined in this way
  - ❑ AWSVPCB.VPC.CREATE - This script creates your VPC, Internet Gateway, route tables, subnets, security groups, S3 buckets, NAT instance, BASTION server, etc. and registers all AWS unique IDs
    - You should only need to run this script once. If you need to start from scratch, you need to run AWSVPCB.VPC.DESTROY before re-running this.
  - ❑ AWSVPCB.VPC.DESTROY - May never be needed - runs VPC.REGISTER to ensure AWS is in synch with the AWSVPCB registry and then destroys your VPC and everything in it
  - ❑ AWSVPCB.VPC.REGISTER - May never be needed. It will find your AWS-VPCB VPC in AWS and register all its components for the other scripts to be able to work as expected - run automatically by VPC.CREATE
  - ❑ AWSVPCB.DIAGLOG - May never be needed. It will upload your environment to an S3 bucket for professor review in case of an issue
- ▶ 4 multi-use assignment scripts as follows:
  - ❑ AWSVPCB.ASSIGNMENT.CREATE # - Destroys existing instances and ELB targets if any exist, then Adjusts VPC settings, Creates assignment instances and ELB (if applicable) based on number passed in as parameter; **Destroys any existing work you've done on assignment.**
  - ❑ AWSVPCB.ASSIGNMENT.START - Starts instances, Creates Route 53 Private zone and resource records; Can run multiple times without destroying work.
  - ❑ AWSVPCB.ASSIGNMENT.STOP - Stops instances, Deletes Route 53 Private zone; Can run multiple times without destroying work.
  - ❑ AWSVPCB.ASSIGNMENT.DESTROY - May never be needed - Destroys existing ELB and instances
    - Called by AWSVPCB.ASSIGNMENT.CREATE; **Destroys all the work you've done on assignment.**

# AWSVPCB Automation & Credits

**Credit Usage Implications of Scripts:** The below assumes you are using the AWS Academy Learner Lab.

- ▶ AWSVPCB.VPC.CREATE - Instances started when first run, but once assignment is stopped, this represents no charges.
- ▶ AWSVPCB.ASSIGNMENT.CREATE - Instances started when first run, but once assignment is stopped, this accounts for charges only in later assignments that use the ELB - a few cents will begin to be billed to your account on a daily basis.
- ▶ AWSVPCB.ASSIGNMENT.START - You will not be able to use your services until you run this script. After you run this script, you will begin to draw down on your credits at a much steeper rate (several dollars per day). The course is designed to allow you hundreds of hours of time, but not 24x7 time for days. **If you do not run the AWSVPCB.ASSIGNMENT.STOP script, whenever you are done working on your assignment, you will run out of credits. Ending the AWS Academy Lab alone does not stop all services!**
- ▶ AWSVPCB.ASSIGNMENT.STOP - will stop the necessary services so that you are no longer using a heavy number of credits while saving all of your changes. While your changes are saved, you will not be able to work until you perform an ASSIGNMENT.START again.
- ▶ AWSVPCB.VPC.DESTROY - If you find that you need to destroy your VPC for a while to save on credits by running this script, that is fine, but please note that you will lose any customizations made to your Bastion server.

# Clouducate Components & Interdependencies

## AWS Virtual Private Cloud Builder (AWSVPCB)

- Collection of BASH shell scripts that take as input json files as definitions for a VPC and assignments. Capabilities include:
  - ❑ Creation of subnets
  - ❑ Creation of security groups with associated rules
  - ❑ Creation of instances
  - ❑ Creation of Load Balancers with associated definitions (e.g. targets, health checks, etc.)
  - ❑ Creation of DNS zone with associated DNS entries
  - ❑ Creation of NAT instance for Internet access
  - ❑ Creation of Client VPN Endpoint for private access over the Internet
  - ❑ Userdata files with instance startup scripts

Domain Name Parameter  
must match

HC AMI (images)  
In AWSVPCB json

Instance & DB PWDs  
In AWSVPCB Directory

## Havoc Circus (HC) Images

- Custom AMIs with pre-built Linux & Windows web applications
- Domain names defined in web configuration files to access DB and define domain in web servers
- Hardcoded admin and database passwords

CA & Domain Certs  
In AWSVPCB Directory

VPC & Assignment json  
In AWSVPCB Dir or S3 Bucket

Logs & Diag Info  
To professor S3 Bucket

# Sample AWSVPCB Configuration Files

## VPC json - in secfiles/vpc#

```
{
  "AWSVPCB": {
    "VPC": {
      {
        "VPCIDR": "172.31.0.0/16"
      }
    },
    "Subnets": [
      {
        "SubnetName": "DEFAULT",
        "SubnetCIDR": "172.31.131.0/24",
        "SecurityGroup": "yes",
        "RoutingTable": "DEFAULT"
      },
      .....
    ],
    "PossibleInstanceNames": [
      {
        "InstanceName": "IIS1"
      },
      .....
    ],
    "PossibleELBs": [
      {
        "ELBName": "myfiu"
      },
      .....
    ],
    "BASTIONDefinition": {
      {
        "IPAddress": "172.31.132.20",
        "Subnet": "PUBLIC",
        "AMI": "ami-041bd92954e73c009"
      },
      {
        "RemoteConnFlag": "bastion"
      },
      .....
    ],
    "NATDefinition": {
      {
        "IPAddress": "172.31.132.151",
        "Subnet": "PUBLIC",
        "AMI": "ami-00d1f8201864cc10c"
      },
      {
        "ServerPrivateKey": "privkey"
      }
    }
  }
}
```

## Assignment jsons - in secfiles/assignment#

```
{
  "AWSVPCB": {
    "Instances": [
      {
        "InstanceName": "LINUX1",
        "InstanceIP": "172.31.128.43",
        "InstanceSubnet": "WEB",
        "InstanceAMI": "ami-0fb4c2197eba775de",
        "InstanceType": "t2.micro"
      },
      .....
    ],
    "FirewallRules": [
      {
        "SecurityGroup": "DEFAULT",
        "RuleType": "inbound",
        "Protocol": "all",
        "Port": "all",
        "SourceGroup": "172.31.0.0/16"
      },
      .....
    ],
    "DNSEntriesFile": "assignment9/awsvpcb.assignment9.DNS.json",
    {
      {
        "ELBs": [
          {
            "ELBName": "rainforest",
            "ListenerProtocol": "HTTPS",
            "ListenerPort": "443",
            "InstanceProtocol": "HTTP",
            "InstancePort": "80",
            "ELBSubnet": "AWSVPCB_ELB",
            "HealthCheckTarget": "TCP:80",
            "HealthCheckInterval": "5",
            "HealthCheckTimeout": "3",
            "HealthCheckUnhealthyThreshold": "2",
            "HealthCheckHealthyThreshold": "2",
            "ELBInstances": [
              {
                "InstanceName": "LINUX1"
              },
              {
                "InstanceName": "LINUX2"
              }
            ]
          },
          .....
        ]
      }
    }
  }
}
```

## vpcb-config - in main directory

```
export INSTITUTION="FIU"
export PROFESSOR="Name"
export COURSE=DYNAMIC
export AVAILABLE_COURSES="... ..."
export AWSVPCB_SEM=DYNAMIC
export ERRORMSG="please run
./AWSVPCB.DIAGLOG and let the
professor know to review...."

export DOMAIN=awsvpcb.edu
export AWS_ACADEMY=yes
export AWS_REGION=us-east-1
export PRIMARY_AWS_AZ=us-east-1a

export
DIAGLOG_LAMBDA_URL=https://....
# Unique per professor
```

# AWSVPCB Miscellaneous Features/Limitations

## FEATURES:

- ▶ Automatic creation of OVPN file for VPN connectivity
- ▶ Automatic creating of DNS records for ELBs
- ▶ Automatic assignment of SSL certs to ELBs, but files (.cert & .key) with ELB names must exist in secfiles directory
- ▶ Automatic saving of any changes to DNS or Firewall rules in between START and STOP scripts; NOTE: ELB(s) are created with the first START and not deleted until the ASSIGNMENT DESTROY
- ▶ Ability to copy all VPC information to S3 bucket (DIAGLOG script), but professor must run aws-setup to create necessary resources in their account

## LIMITATIONS:

- ▶ At this time json file syntax validation is non-existent; this is in the backlog to be built
  - ❑ Mitigation offered through MANIFEST.DISPLAY script, but it still requires manual verification
- ▶ Security groups are only created with association to subnet. Thus, two subnets cannot share the same security group
- ▶ Only a subset of parameters are accepted for each AWS instance
- ▶ By default, AWS limits the number of SSL certificate uploads, so too many VPC creations will cause errors - try to limit the number of VPC creations - there is no limit to number of assignment creations
- ▶ Active VPC entities are stored in files within the AWSVPCB directory, so you must run the scripts from the same directory
  - ❑ Portability to another system offered through CONFIGURE and VPC.REGISTER scripts

# AWSVPCB - How to start quickly #1

1. USE awsvpcb-setup (instructions here - <https://github.com/clouducate/awsvpcb/raw/refs/heads/main/awsvpcb-setup%20instructions.docx>)
    - ▶ NOTE: This requires you to create an AWS account with a credit card provided for AWS, however, expenses are minimal (less than \$1 per month)
    - ▶ NOTE: By default the awsvpcb-setup script will send logs and resulting vpcb-config file generated to centralized admin S3 bucket, but you can override this when prompted whether to add to registry or not as indicated in the instructions.
    - ▶ NOTE: Your login credentials expire after 12 hours, so you will need to re-login at that point using the "aws login" command
  2. From the same device as used in step #1, perform the following:
    - ▶ TYPE: wget <https://github.com/clouducate/awsvpcb/raw/refs/heads/main/awsvpcb-scripts.zip>
    - ▶ TYPE: unzip awsvpcb-scripts.zip
    - ▶ TYPE: cd awsvpcb-scripts
    - ▶ TYPE: chmod 755 \*
    - ▶ TYPE: chmod 755 procs/\*
  3. COPY the vpcb-config file created in step #1 to this directory then do the following:
    - ▶ TYPE: ./AWSVPCB.CONFIGURE # If you have multiple course, then you will be asked to choose one
    - ▶ TYPE: ./AWSVPCB.VPC.CREATE # Up to this point there are no cost implications to what's been done
    - ▶ TYPE: ./AWSVPCB.ASSIGNMENT.CREATE 2 # Answer "y" at the prompt - NOTE: costs start here, but it's negligible for a few hours.
- ▶ **AT THIS POINT YOU HAVE CREATED A FULL AWS ENVIRONMENT WITH EC2 INSTANCES, SUBNETS, ROUTE TABLES, LOAD BALANCERS, A DATABASE, ETC. FEEL FREE TO PERUSE IT FROM THE AWS CONSOLE. IF YOU WISH TO LOG INTO YOUR VPC AND TEST AN APPLICATION, YOU CAN DO THE BELOW. IF NOT SKIP TO STEP 15.**

# AWSVPCB - How to start quickly #2

4. Log into the AWS console. Place your cursor on the “Search” box and type: ec2
5. Click on the “EC2” selection in the drop down. Scroll down on the left-side pane until you see “Security Groups” and click on it.
6. Click on the checkbox next to the “AWS-VPCB-PUBLIC” security group. When you do, the bottom portion of the screen will populate. Click on “inbound rules” and then on the “Edit inbound rules” button.
7. You should be presented with the below screen. Click on the “Add rule” button. Then, click on the “Custom TCP” button and type RDP to select it. Then, click on the “Custom” button and select “My IP” from the drop-down menu. Finally, click on the “Save rules” button.
8. Scroll up on the left-side pane until you see “Instances” and click on it. Then click on the checkbox next to “AWS-VPCB-BASTION”. This will populate the bottom part of the page with the details about this server. Click on the boxes under “Public IPv4 DNS” to copy the public name of the server.
9. Open a Remote Desktop Tool
  - ▶ ON MAC - Download Microsoft Remote Desktop 10 or higher from the MAC App Store and install.
  - ▶ ON WINDOWS - There are several Remote Desktop options, but “Remote Desktop Connection” comes pre-installed on all Windows machines. You simply need to “rdp” on the Windows search bar. However, you will need to constantly re-enter your credentials each time you connect.
10. Now, let’s connect to your Bastion host! Within RDP, click on “Show options” to expand the window. Paste in the DNS name you just copied in the “Computer” field. Type “administrator” in the “User name” field. Finally, click on the “Connect” button.
11. You may be asked “Do you trust this remote connection?”. Simply click “Connect” again.
12. You should then be prompted for the password. Copy and paste the following as the password:  
P@oDxV)4-nPUfp\$Ar?V@N9Lpjbsnp@W!



# AWSVPCB - How to start quickly #3

13. You will then be prompted again with something like “The identity of the remote computer cannot be verified. Do you want to connect anyway?”. Click “yes”. You are now logged into your Bastion host!
14. To test the applications associated to this particular assignment number, simply open a browser on your Bastion server and type [HTTP://myfiu.awsvpcb.edu](http://myfiu.awsvpcb.edu)

**AT THIS POINT YOU ARE IN YOUR VPC AND LOOGED INTO A WINDOWS SERVER WITH MANY TOOLS PRE-INSTALLED. FEEL FREE TO LOOK AROUND.**

15. Once you are done testing, you should destroy the assignment to avoid AWS costs. To do so, simply go back to the awsvpcb-scripts directory and run `./AWSVPCB.ASSIGNMNET.DESTROY`.
16. You can experiment with other assignments and/or destroy it all whenever you wish.
  - ▶ NOTE: Assignment numbers beyond 3 have purposefully created errors and the applications ([HTTP://myfiu.awsvpcb.edu](http://myfiu.awsvpcb.edu) and [HTTP://rainforest.awsvpcb.edu](http://rainforest.awsvpcb.edu)) will not work as expected.

# Next Steps for your course

## MAP OUT TARGET ENVIRONMENT

- ▶ What AMIs do you plan to use? Are the default AMIs provided sufficient, or do you want to take those and create custom AMIs?
- ▶ What applications will the students be testing? Are the default rainforest and myfiu applications sufficient or do I want to build/use others?
- ▶ What do you want your VPC to look like (IP range, subnets, possible instances, possible ELBs)? Is the default sufficient?
- ▶ What do you want your assignments to look like (instances, firewall rules, DNS entries, ELBs)? Are the default assignments sufficient?
- ▶ What do you want your assignments to do (e.g. setup a security vulnerability or break the environment in some way)?
- ▶ Use default awsvpcb.edu domain OR create your own domain and provide the following:
  - ❑ Public CA cert for import into student client machines or use public CA (e.g., Sectigo)
  - ❑ Client & Server VPN certs (if using VPN - not recommended) and private keys
  - ❑ ELB certs and private keys
- ▶ Use pre-defined VPC and assignment AWSVPCB json files OR create your own
- ▶ Use the AWS Academy Learner Lab for the students (recommended) OR each student needs to create AWS account (with credit card) and install the AWS CLI; the awsvpcb-student-setup script with the following instructions (<https://github.com/clouducate/awsvpcb/raw/refs/heads/main/awsvpcb-student-setup%20instructions.docx>) can also be provided.

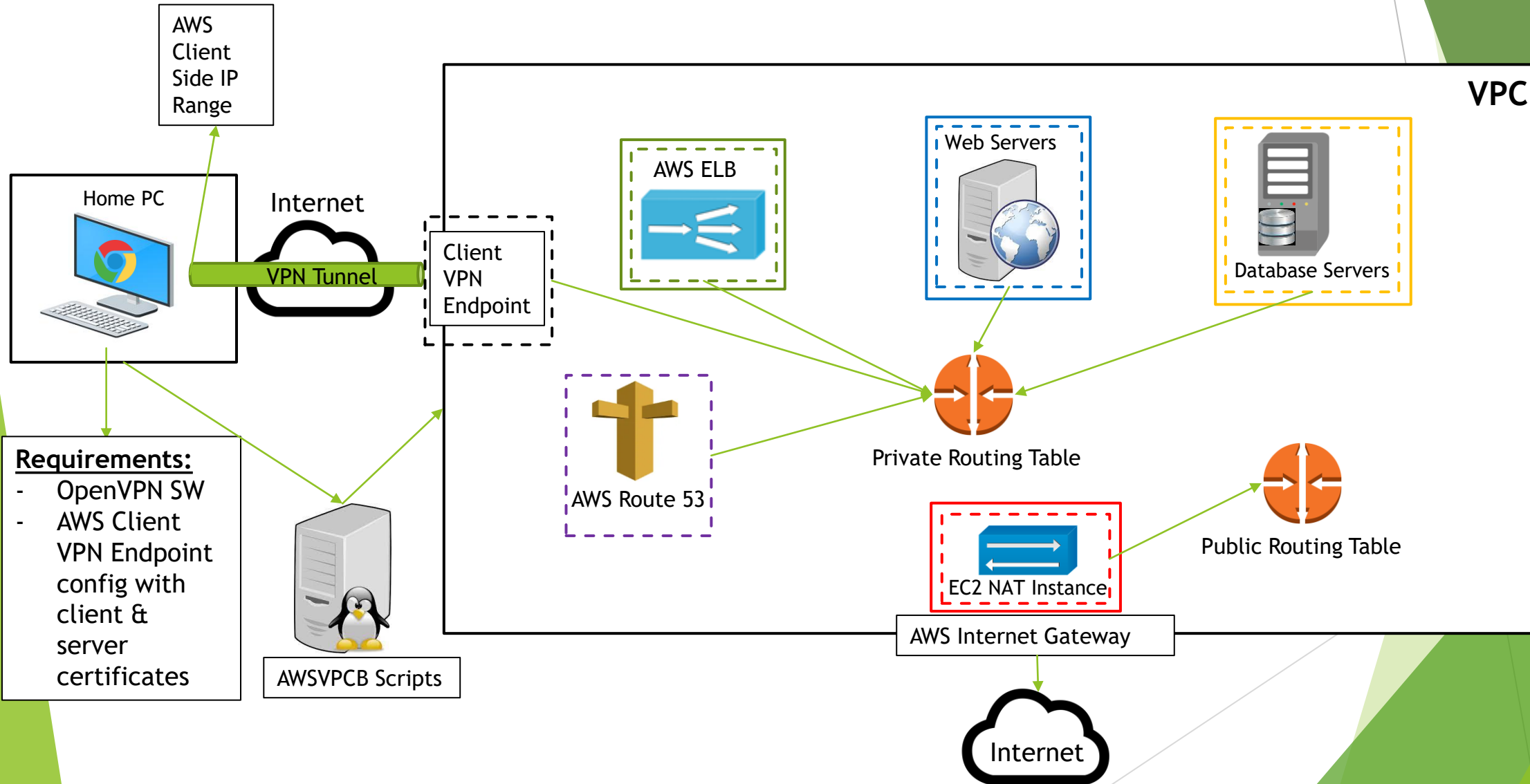
# Appendix

Older features (e.g., VPN, Havoc Circus Windows Service, Cloudwatch Logging) that are still supported, but not actively used

# AWSVPCB Legacy Config (vpc1)

Legend:

— Security Group  
- - - Subnet



# Legacy Clouducate Components & Interdependencies

## AWS Virtual Private Cloud Builder (AWSVPCB)

- Collection of BASH shell scripts that take as input json files as definitions for a VPC and assignments. Capabilities include:
  - ❑ Creation of subnets
  - ❑ Creation of security groups with associated rules
  - ❑ Creation of instances
  - ❑ Creation of Load Balancers with associated definitions (e.g. targets, health checks, etc.)
  - ❑ Creation of DNS zone with associated DNS entries
  - ❑ Creation of NAT instance for Internet access
  - ❑ Creation of Client VPN Endpoint for private access over the Internet

CA & Domain Certs  
In AWSVPCB Directory

VPC & Assignment json  
In AWSVPCB Dir or S3 Bucket

Domain Name Parameter  
must match

HC AMI (images)  
In AWSVPCB json

AWSVPCB DNS IPs  
In HC AMIs

AWSVPCB Assignment #  
In DNS TXT Record

Instance & DB PWDs  
In AWSVPCB Directory

Logs & Diag Info  
To S3 Bucket

## Havoc Circus (HC)

- Windows Service written in C-Sharp that takes as input json files as definitions for modifications for assignments. Capabilities include:
  - ❑ Custom AMIs with pre-built Linux & Windows web applications
  - ❑ Reading of assignment # from DNS TXT record
  - ❑ Running a remote command on any system within the VPC. In this way just about any problem can be created dynamically
  - ❑ At shutdown, verifying whether a problem was addressed and if so, marking it as complete to avoid restarting a solved problem

Assignment files  
In S3 Bucket

Assignment json  
In S3 Bucket

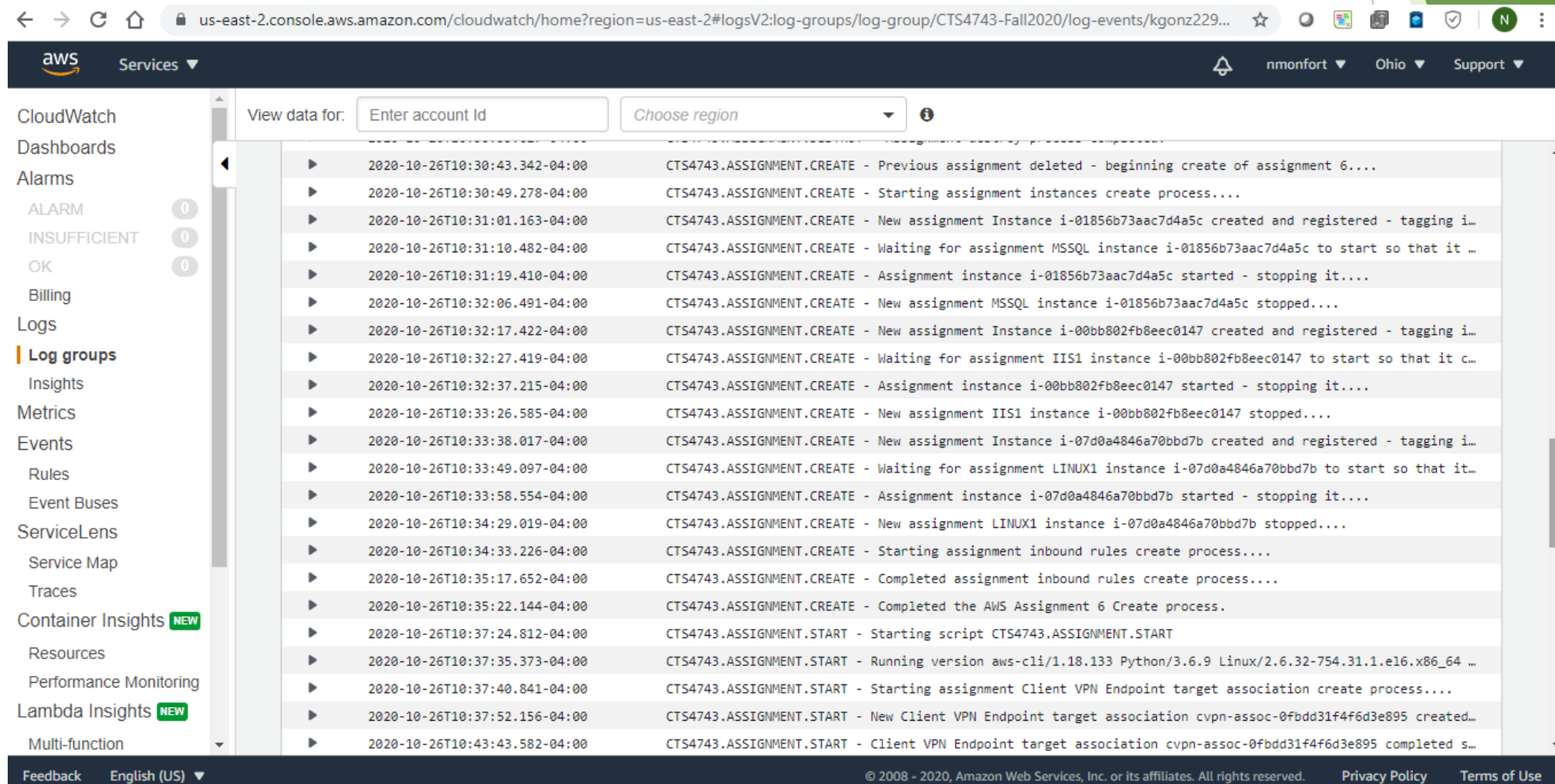
# Sample Havoc Circus json Files

```
{
  "HavocMonkey": {
    "PrimateAssembly": {
      "PrimateTypeName": "Assignment5.HavocMonkey",
      "PrimateFileName": "Assignment5.HavocMonkey.dll",
      "PrimateDependencies": [
        {
          "DependencyFileName": "Primate.dll"
        }
      ]
    },
    "PrimatePackages": [
      {
        "PrimatePackage": {
          "TargetHostName": "iis1.cts4743.edu",
          "TargetUserName": "Administrator",
          "TargetUserPassword": "P@oDxV)4-nPUfp$Ar?V@N9Lpjbsnp@W!",
          "TargetPlatform": "Windows",
          "TargetPackageFiles": [
            {
              "FileName": "MemoryLocker.exe",
              "FilePath": "C:\\Apps\\MemoryLocker\\"
            }
          ]
        },
        {
          "FileName": "MemoryWorker.runtimeconfig.json",
          "FilePath": "C:\\Apps\\MemoryWorker\\"
        }
      ],
      {
        "PrimatePackage": {
          "TargetHostName": "linux1.cts4743.edu",
          "TargetUserName": "ec2-user",
          "TargetUserPassword": "cts4743",
          "TargetPlatform": "Linux",
          "TargetPackageFiles": [
            {
              "FileName": "CPUWorker.exe",
              "FilePath": "/usr/bin/cts4743/apps/cpu_worker/"
            }
          ]
        },
        {
          "FileName": "CPUWorker.runtimeconfig.json",
          "FilePath": "/usr/bin/cts4743/apps/cpu_worker/"
        }
      ]
    ],
    "PrimateTasks": [
      {
        "PrimateTask": {
          "TaskLabel": "MemoryWorker",
          "TargetHostName": "iis1.cts4743.edu",
          "TargetUserName": "Administrator",
          "TargetUserPassword": "P@oDxV)4-nPUfp$Ar?V@N9Lpjbsnp@W!",
          "TargetPlatform": "Windows",
          "TaskInitializationCommands": [
            {
              "CommandExecutable": "C:\\Apps\\MemoryLocker\\MemoryLocker.exe",
              "CommandArguments": "384"
            }
          ]
        }
      },
      {
        "PrimateTask": {
          "TaskLabel": "CPUWorker",
          "TargetHostName": "linux1.cts4743.edu",
          "TargetUserName": "ec2-user",
          "TargetUserPassword": "cts4743",
          "TargetPlatform": "Linux",
          "TaskInitializationCommands": [
            {
              "CommandExecutable": "nohup dotnet \"/usr/bin/cts4743/apps/cpu_worker/CPUWorker.dll 1 20000000000 125\" > /dev/null 2>&1 & \\",
              "CommandArguments": ""
            }
          ]
        }
      ]
    ]
  }
}
```

```
{
  "HavocMonkey": {
    "PrimateAssembly": {
      "PrimateTypeName": "Assignment8.HavocMonkey",
      "PrimateFileName": "Assignment8.HavocMonkey.dll",
      "PrimateDependencies": [
        {
          "DependencyFileName": "Primate.dll"
        }
      ]
    },
    "PrimatePackages": [
      {
        "PrimatePackage": {
          "TargetHostName": "iis1.cts4743.edu",
          "TargetUserName": "Administrator",
          "TargetUserPassword": "P@oDxV)4-nPUfp$Ar?V@N9Lpjbsnp@W!",
          "TargetPlatform": "Windows",
          "TargetPackageFiles": [
            {
              "FileName": "Web.config",
              "FilePath": "C:\\Web\\MyFIU\\",
              "ForceOverwrite": "true"
            }
          ]
        }
      ]
    ],
    "PrimateTasks": [
      {
        "PrimateTask": {
          "TaskLabel": "IISWebConfigMangler",
          "TargetHostName": "iis1.cts4743.edu",
          "TargetUserName": "Administrator",
          "TargetUserPassword": "P@oDxV)4-nPUfp$Ar?V@N9Lpjbsnp@W!",
          "TargetPlatform": "Windows",
          "TaskInitializationCommands": [
            {
              "CommandExecutable": "C:\\Windows\\System32\\cmd.exe",
              "CommandArguments": "/C echo Assignment 8 initialization success on %DATE% %TIME%! > C:\\Assignment8.txt"
            }
          ]
        }
      ]
    ]
  }
}
```

# AWSVPCB Logging

- Optional logging to AWS cloudwatch is built into the AWSVPCB system. This allows you to track the progress of students and whether they made a mistake when running the scripts. AWS account with IAM user that has access to cloudwatch must be created outside of tool.



The screenshot displays the AWS CloudWatch console interface. The top navigation bar shows the AWS logo, 'Services' dropdown, and user information 'nmonfort' in 'Ohio'. The left sidebar contains a menu with options like CloudWatch, Dashboards, Alarms, Logs, and Log groups. The 'Log groups' section is expanded, showing a list of log groups. The main content area displays a table of log events for the selected log group. The table has three columns: a timestamp, a log type, and a log message. The log messages describe the creation and configuration of assignment instances and rules.

Timestamp	Log Type	Log Message
2020-10-26T10:30:43.342-04:00	CTS4743.ASSIGNMENT.CREATE	- Previous assignment deleted - beginning create of assignment 6...
2020-10-26T10:30:49.278-04:00	CTS4743.ASSIGNMENT.CREATE	- Starting assignment instances create process...
2020-10-26T10:31:01.163-04:00	CTS4743.ASSIGNMENT.CREATE	- New assignment Instance i-01856b73aac7d4a5c created and registered - tagging i...
2020-10-26T10:31:10.482-04:00	CTS4743.ASSIGNMENT.CREATE	- Waiting for assignment MSSQL instance i-01856b73aac7d4a5c to start so that it ...
2020-10-26T10:31:19.410-04:00	CTS4743.ASSIGNMENT.CREATE	- Assignment instance i-01856b73aac7d4a5c started - stopping it...
2020-10-26T10:32:06.491-04:00	CTS4743.ASSIGNMENT.CREATE	- New assignment MSSQL instance i-01856b73aac7d4a5c stopped...
2020-10-26T10:32:17.422-04:00	CTS4743.ASSIGNMENT.CREATE	- New assignment Instance i-00bb802fb8eec0147 created and registered - tagging i...
2020-10-26T10:32:27.419-04:00	CTS4743.ASSIGNMENT.CREATE	- Waiting for assignment IIS1 instance i-00bb802fb8eec0147 to start so that it c...
2020-10-26T10:32:37.215-04:00	CTS4743.ASSIGNMENT.CREATE	- Assignment instance i-00bb802fb8eec0147 started - stopping it...
2020-10-26T10:33:26.585-04:00	CTS4743.ASSIGNMENT.CREATE	- New assignment IIS1 instance i-00bb802fb8eec0147 stopped...
2020-10-26T10:33:38.017-04:00	CTS4743.ASSIGNMENT.CREATE	- New assignment Instance i-07d0a4846a70bbd7b created and registered - tagging i...
2020-10-26T10:33:49.097-04:00	CTS4743.ASSIGNMENT.CREATE	- Waiting for assignment LINUX1 instance i-07d0a4846a70bbd7b to start so that it...
2020-10-26T10:33:58.554-04:00	CTS4743.ASSIGNMENT.CREATE	- Assignment instance i-07d0a4846a70bbd7b started - stopping it...
2020-10-26T10:34:29.019-04:00	CTS4743.ASSIGNMENT.CREATE	- New assignment LINUX1 instance i-07d0a4846a70bbd7b stopped...
2020-10-26T10:34:33.226-04:00	CTS4743.ASSIGNMENT.CREATE	- Starting assignment inbound rules create process....
2020-10-26T10:35:17.652-04:00	CTS4743.ASSIGNMENT.CREATE	- Completed assignment inbound rules create process....
2020-10-26T10:35:22.144-04:00	CTS4743.ASSIGNMENT.CREATE	- Completed the AWS Assignment 6 Create process.
2020-10-26T10:37:24.812-04:00	CTS4743.ASSIGNMENT.START	- Starting script CTS4743.ASSIGNMENT.START
2020-10-26T10:37:35.373-04:00	CTS4743.ASSIGNMENT.START	- Running version aws-cli/1.18.133 Python/3.6.9 Linux/2.6.32-754.31.1.el6.x86_64 ...
2020-10-26T10:37:40.841-04:00	CTS4743.ASSIGNMENT.START	- Starting assignment Client VPN Endpoint target association create process....
2020-10-26T10:37:52.156-04:00	CTS4743.ASSIGNMENT.START	- New Client VPN Endpoint target association cvpn-assoc-0fbdd31f4f6d3e895 created...
2020-10-26T10:43:43.582-04:00	CTS4743.ASSIGNMENT.START	- Client VPN Endpoint target association cvpn-assoc-0fbdd31f4f6d3e895 completed s...

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use