

The background features abstract, overlapping green geometric shapes in various shades, creating a modern and dynamic feel. The shapes are primarily triangular and polygonal, with some areas being more opaque than others, creating a layered effect.

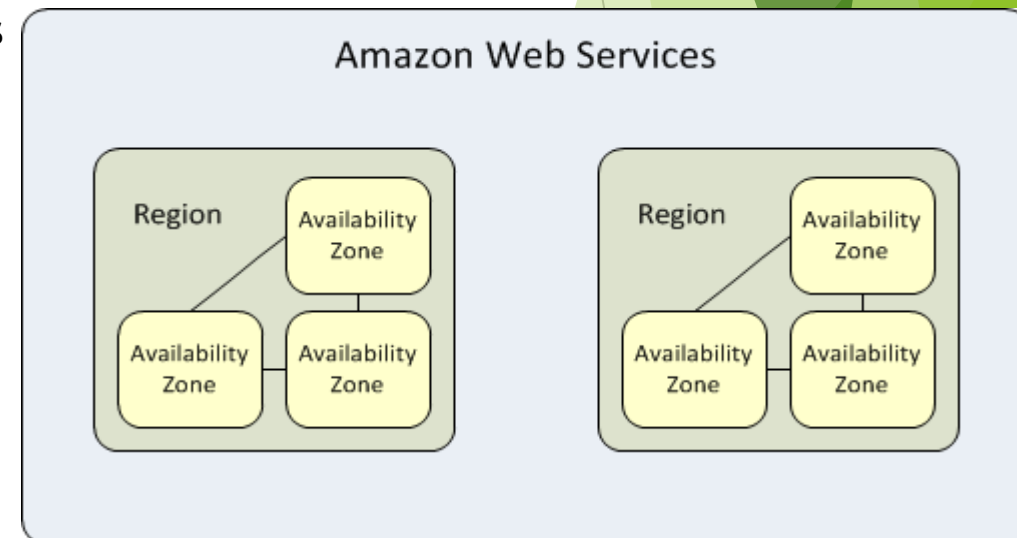
Clouducate

Enabling Cloud-based IT Hands-on Education

Norbert Monfort & Robert Fortunato

AWS Concepts & Glossary

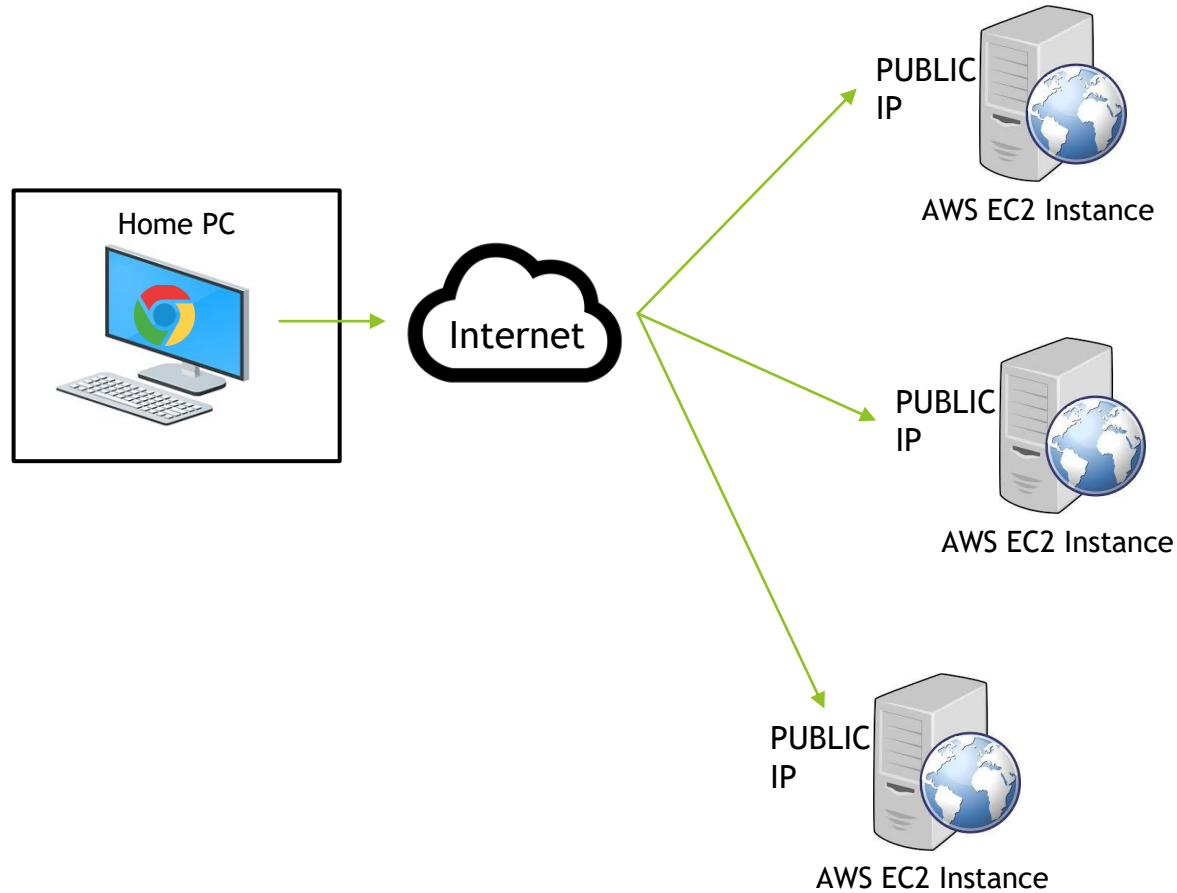
- ▶ AWS Educate - AWS Offering (FIU enrolled)
 - ❑ For this class, I have obtained AWS credits that I will distribute (\$100 per student) - there is no need to enroll in AWS Educate.
- ▶ AWS Region - One or more AWS data centers within a nearby area (within 250 miles) - We will be using “us-east-2”
- ▶ AWS Availability zone - One datacenter within a region - We will be using 1 AZ in this course - “us-east-2A”
- ▶ AWS VPC (Virtual Private Cloud) - Private network within AWS includes large private address space; can span Availability zones, but not regions
 - ❑ We will be using this as it simulates an Enterprise environment
- ▶ AWS VPC Subnet - Portion of VPC addresses used to group instances & services; Cannot span availability zones
- ▶ IAM User - Need to create user (any name) to administrate your services as opposed to account login
- ▶ User Access Key ID and Access Secret Key (associated to IAM user) - Needed for AWS CLI (command line interface) scripts
- ▶ S3 Storage - AWS’ Simple Storage Service (object storage via Internet access)
 - ❑ We will be using this service for Load Balancer logs



AWS Concepts & Glossary

- ▶ EC2 (Elastic Compute Cloud) Instance - A VM server in AWS (can be associated to a VPC subnet and a security group)
- ▶ 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
- ▶ Internet Gateway - Allows Internet access out of VPC
- ▶ Client VPN Endpoint - Allows VPN access into VPC - Associated to a VPC subnet
- ▶ Route 53 - AWS' DNS service - Associated to VPC IP addresses (inbound endpoints)
 - ❑ Can include 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



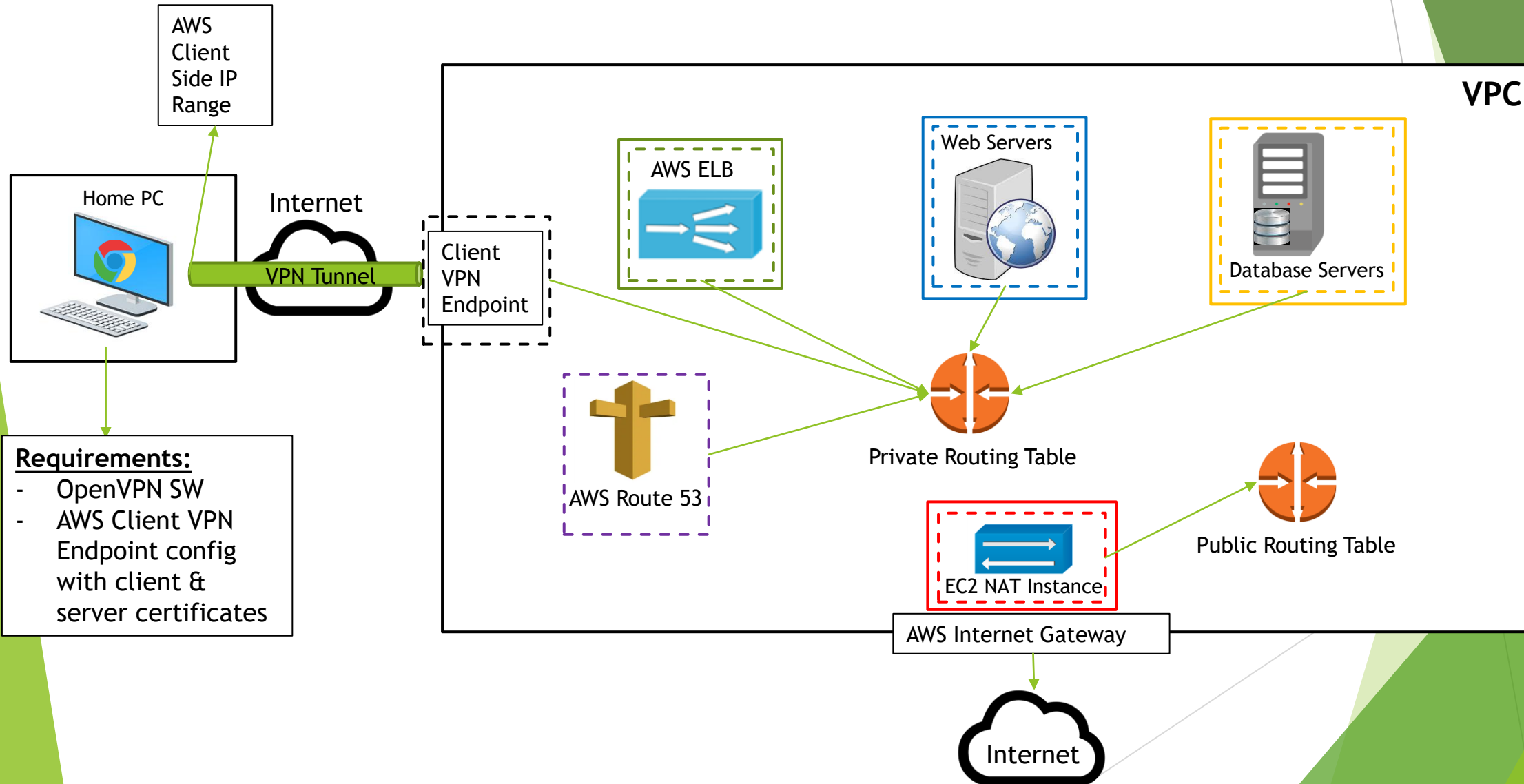
► Problems with this config:

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

AWS CTS-4743 VPC Config

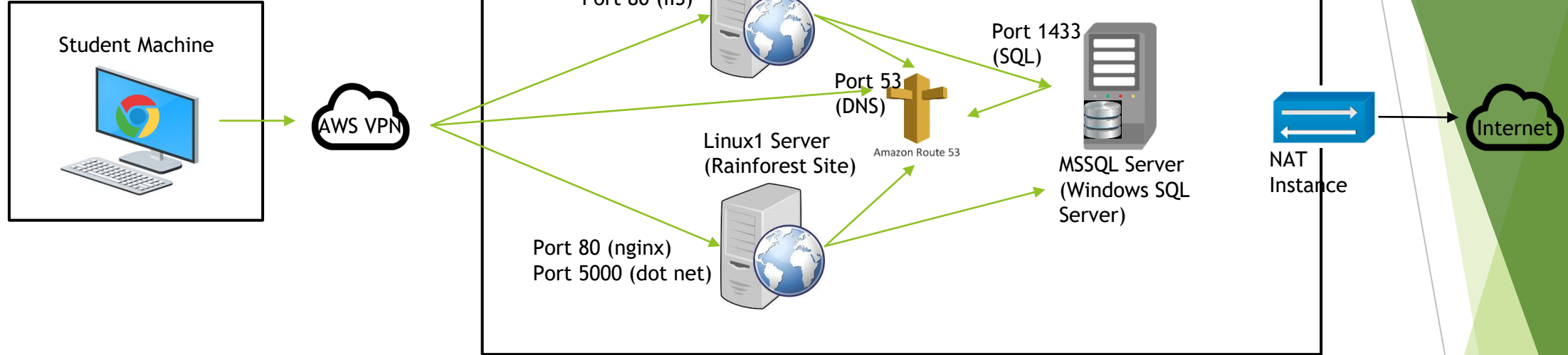
Legend:

— Security Group
- - - Subnet

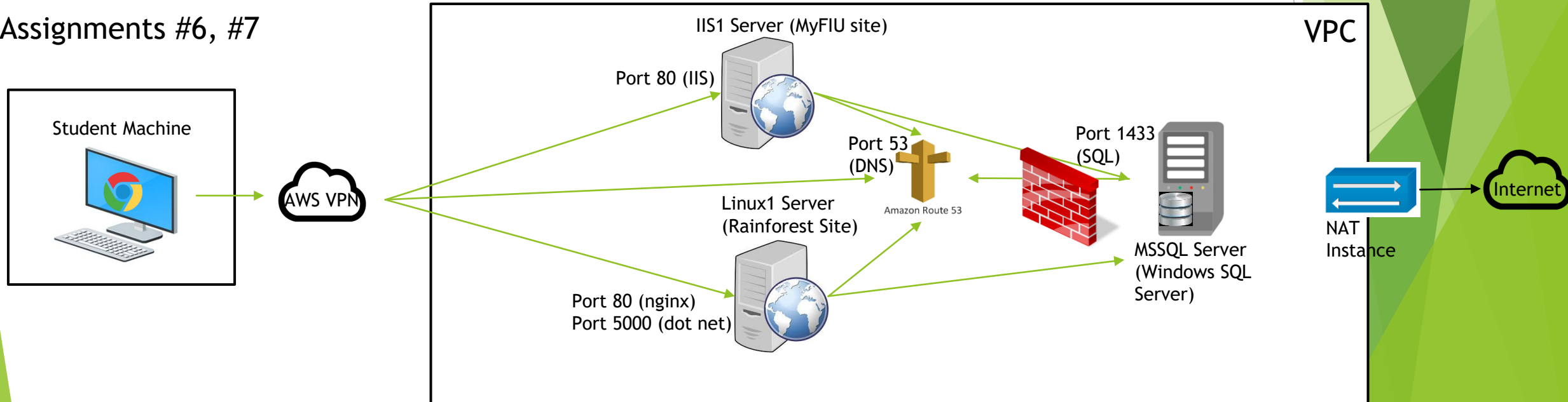


AWS CTS-4743 Assignment Diagrams

Assignments #3, #4, #5

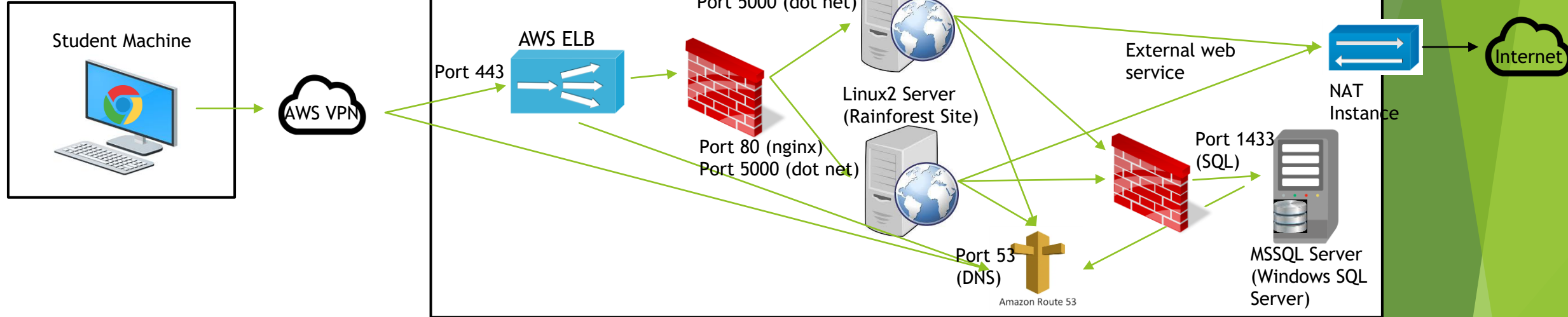


Assignments #6, #7

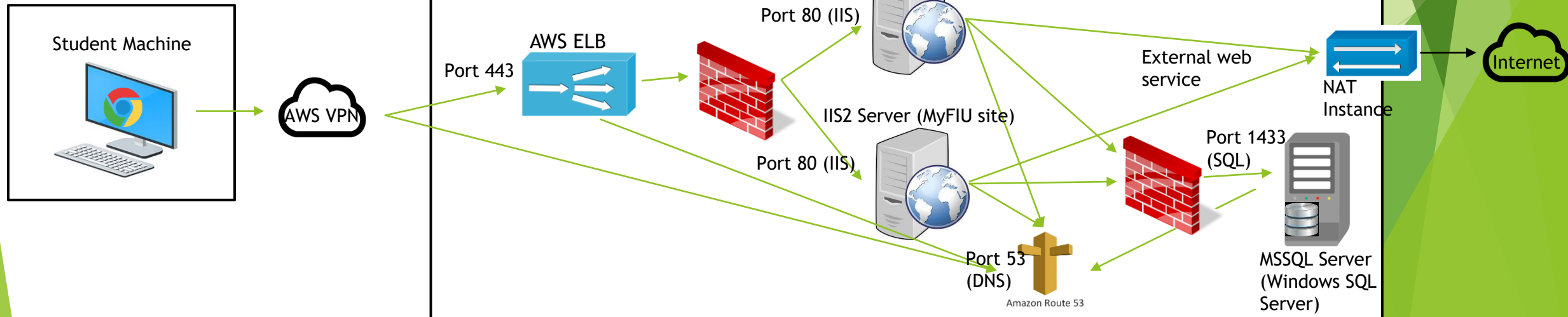


AWS CTS-4743 Assignment Diagrams

Assignments #9, #11 Setup #1 - Linux



Assignments #8, #10 Setup #2 - Windows



AWSVPCB SCRIPTS

- ▶ In this course you will be provided with several scripts to manage your VPC
- ▶ 5 one-time scripts as follows and 1 recovery script:
 - ❑ AWSVPCB.CONFIGURE & AWSVPCB.TEST - Only needed at start of course, but can be run again
 - ❑ AWSVPCB.VPC.CREATE - This script creates your VPC, Internet Gateway, route tables, subnets, security groups, Route 53 zone with record entries and Client VPN endpoint and registers all AWS unique IDs
 - This script will fail if a “AWS-VPCB” tagged VPC exists
 - You should only need to run this script once. If you need to start from scratch, you should run AWSVPCB.VPC.DESTROY before re-running this.
 - ❑ AWSVPCB.VPC.DESTROY - This script will destroy the registered VPC and everything in it
 - You should only need to run this at the end of the course; running this will require replacing VPN config
 - ❑ AWSVPCB.VPC.REGISTER - This script 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
- ▶ 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, Associates Client VPN endpoint to subnet, Creates Route 53 Inbound endpoints; Can run multiple times without destroying work.
 - ❑ AWSVPCB.ASSIGNMENT.STOP - Stops instances, Disassociate Client VPN endpoint from subnet, Deletes Route 53 Inbound Endpoints; Can run multiple times without destroying work.
 - ❑ AWSVPCB.ASSIGNMENT.DESTROY - 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 a new AWS account. If you have an account older than 12 months will be charged a small amount after VPC.CREATE and a little more after ASSIGNMENT.CREATE. If you are in this position, be more mindful of your spend status, but should still have enough credits for the course.

- ▶ AWSVPCB.VPC.CREATE - No charges after this script is run.
- ▶ AWSVPCB.ASSIGNMENT.CREATE - After running this script for 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.**
- ▶ AWSVPCB.ASSIGNMENT.STOP - will stop the necessary services so that you are no longer using a heavy amount 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 need to re-import your VPN configuration after recreating your VPC.

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

VPC json

Sample AWSVPCB Configuration Files

Assignment jsons

vpcb-config

```
{
  "AWSVPCB": {
    "VPC": {
      "VPCCIDR": "172.31.0.0/16"
    },
    "Subnets": [
      {
        "SubnetName": "DEFAULT",
        "SubnetCIDR": "172.31.131.0/24",
        "SecurityGroup": "yes",
        "RoutingTable": "DEFAULT"
      }
    ],
    "PossibleInstanceNames": [
      {
        "InstanceName": "IIS1"
      }
    ],
    "PossibleELBs": [
      {
        "ELBName": "myfiu"
      }
    ],
    "DNSIPAddresses": [
      {
        "IPAddress": "172.31.131.10"
      }
    ],
    "NATDefinition": {
      "IPAddress": "172.31.132.151",
      "Subnet": "PUBLIC",
      "AMI": "ami-00d1f8201864cc10c"
    },
    "VPNDefinition": {
      "CACert": "ca.crt",
      "ConfigFile": "AWSVPCB-client-config.ovpn",
      "ClientCIDR": "172.31.8.0/22"
    },
    "ServerPrivateKey": "privkey"
  }
}
```

```
{
  "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": "ELB",
        "HealthCheckTarget": "TCP:80",
        "HealthCheckInterval": "5",
        "HealthCheckTimeout": "3",
        "HealthCheckUnhealthyThreshold": "2",
        "HealthCheckHealthyThreshold": "2",
        "ELBInstances": [
          {
            "InstanceName": "LINUX1"
          },
          {
            "InstanceName": "LINUX2"
          }
        ]
      }
    ]
  }
}
```

```
{
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "nat.awsvpcb.edu",
        "Type": "A",
        "TTL": 300,
        "ResourceRecords": [
          {
            "Value": "172.31.132.151"
          }
        ]
      }
    },
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "linux2.awsvpcb.edu",
        "Type": "A",
        "TTL": 300,
        "ResourceRecords": [
          {
            "Value": "172.31.128.44"
          }
        ]
      }
    },
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "mssql.awsvpcb.edu",
        "Type": "A",
        "TTL": 300,
        "ResourceRecords": [
          {
            "Value": "172.31.129.75"
          }
        ]
      }
    }
  ]
}
```

```
export COURSE=CTS-4743
export SEMESTER=$COURSE-Fall2020
export DOMAIN=awsvpcb.edu
export AWSCMD=aws

export MANIFEST_LOCATION=local
#export MANIFEST_LOCATION=aws

export ENABLE_AWS_LOGGING=yes
#export ENABLE_AWS_LOGGING=no

export AWS_REGION=us-east-2
export AWS_AZ=us-east-2a

export LOGGING_ACCESS_KEY=.....
#Only used if ENABLE_AWS_LOGGING=yes

export LOGGING_SECRET_KEY=.....
#Only used if ENABLE_AWS_LOGGING=yes

export MANIFEST_ACCESS_KEY=.....
#Only used if MANIFEST_LOCATION=aws

export MANIFEST_SECRET_KEY=.....
#Only used if MANIFEST_LOCATION=aws

export MANIFEST_S3BUCKET=

export DIAGLOG_S3BUCKET=
#Requires LOGGING_ACCESS_KEY and
LOGGING_SECRET_KEY

## LOGGING and MANIFEST access and secret
keys can be the same if you use the same
account to house both.
```

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": ""
              }
            ]
          }
        }
      ]
    }
  }
}
```

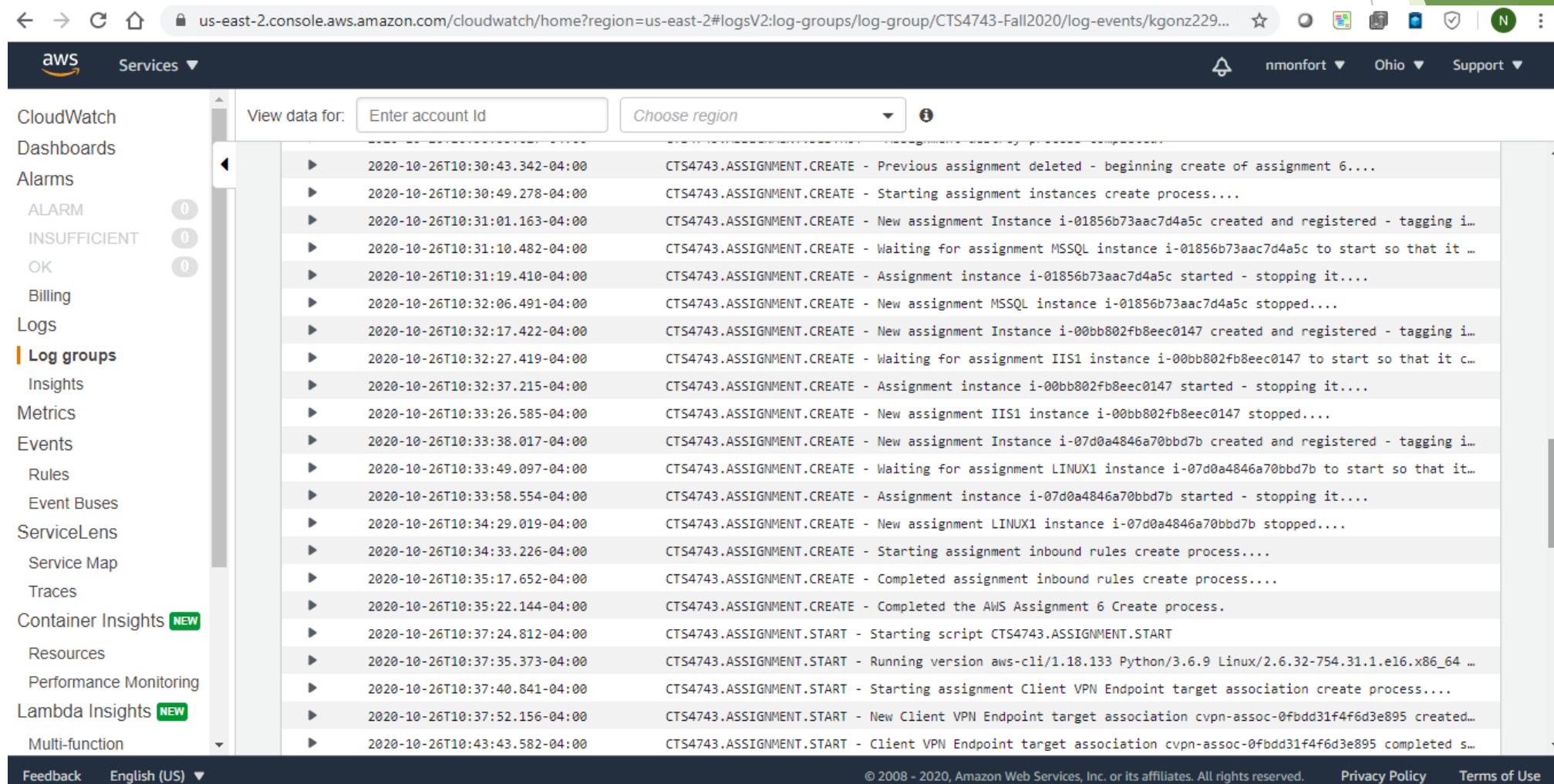
```
{
  "HavocMonkey": {
    "PrimateAssembly": {
      "PrimateTypeName": "Assignment8.HavocMonkey",
      "PrimateFileName": "Assignment8.HavocMonkey.dll",
      "PrimateDependencies": [
        {
          "DependencyFileName": "Primate.dll"
        }
      ]
    },
    "PrimatePackages": [
      {
        "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"
            }
          ]
        }
      ]
    }
  }
}
```

Clouducate Pre-Requisites

- ▶ 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 and private keys
 - ❑ ELB certs and private keys
- ▶ Create one or more AWS accounts for the following purposes:
 - ❑ Optionally, create server AMIs to be used for assignments
 - Havoc Circus AMIs can be used OR publicly available AWS AMIs can be used, but at least one custom AMI is needed for the server that will run Havoc Circus as AWSVPCB DNS servers need to be configured on it.
 - ❑ Maintain any custom AMIs used for the assignments (at least one is needed as noted above)
 - ❑ House Havoc Circus assignment dependency files and json files
 - ❑ Optionally, house AWSVPCB VPC and assignment json files
 - ❑ Optionally, if you wish to enable AWS logging, then IAM user with Cloudwatch access
 - ❑ Optionally, if you wish to be able to gather diagnostic information, then same IAM user needs access to be able to create a new S3 bucket
- ▶ Create VPC and assignment AWSVPCB json files
- ▶ Create Havoc Circus assignment json files
- ▶ Each student needs to create AWS account, be provided AWS credits and preferably a Linux server from which they can run the AWS CLI client
- ▶ Access to AMIs must be granted to students' AWS accounts

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 includes the AWS logo, a 'Services' dropdown, and user information (nmonfort, Ohio, Support). The left sidebar contains a navigation menu with options like CloudWatch, Dashboards, Alarms, Logs, and more. The main content area shows a list of log events for a specific log group. The events are timestamped and include detailed descriptions of the actions being performed, such as creating assignment instances, waiting for instances to start, and starting scripts.

Timestamp	Event 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...

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 bucket must be pre-created within the same account used for logging

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
- ▶ Whenever a new VPC is created, a new OVPN file is created requiring a new import on client machines
- ▶ 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