Screening assignment for a DevOps intern that involves writing code, and deploying it to both a Windows IIS machine and a Linux service behind a load balancer on AWS, encompasses several key skills. These skills include coding, familiarity with AWS services, understanding of Windows and Linux operating systems, and knowledge of web servers and load balancers. Here is a structured approach to designing such a test:

1. Overview of the Task

- **Objective**: The intern will write a simple web application, deploy it on an AWS EC2 instance running Windows Server with Internet Information Services (IIS) for the web server, and then deploy the same application on a Linux EC2 instance. The Linux deployment should be configured to run behind an Elastic Load Balancer (ELB) to manage incoming traffic.
- **Skills Tested**:
- Basic programming or scripting abilities.
- Familiarity with Windows Server and IIS.
- Familiarity with Linux, particularly with deploying and running web services.
- Understanding of AWS EC2, Elastic Load Balancing, and basic AWS networking (VPC, subnets, security groups).
- Ability to work with AWS CLI or AWS Management Console.
- Basic understanding of source control (using Git for versioning and source code management).

2. Pre-requisites

- AWS Account: The candidate should create a free tier AWS account and then create a user with permissions to create and manage SQS queues, Lambda functions, and S3 buckets. **Be Sure that candidate has enabled MFA on the root account and user account as well.**
- Development Tools: Access to a code editor or IDE, Git installed for version control, and basic knowledge of command-line tools.
- Basic template or starter code in a language of their preference or specified by the test (e.g., Python Flask app, Node.js Express app, C# or any language of your choice).

3. Test Details

Part 1: Coding the Application **and the Infrastructure**

- **Task 1**:
  - Write a simple "Hello World" web application.
  - The application should display a custom greeting message that includes the current server time.
  - The application should log each request to a file or standard output, including the timestamp and the requested path.
- **Task 2**:
  - Use any of the IaC Language (Terraform, Cloudformation, Pulumi)
  - Write down the entire infrastructure as code
  - Load Balancer, VPC, Subnets, Route Tables, EC2 Instances **must be created using IaC**.

Part 2: Deploying to Windows IIS

- **Task**: Deploy the application on a Windows Server instance on AWS EC2.
- Create a new EC2 instance using the AWS Management Console or AWS CLI, selecting a Windows Server AMI.
  - Do it manually for once, FINALLY, write the configuration to create all the AWS Resources using IaC (of your choice)
- Configure IIS on the Windows Server to host the web application.
- Access the application from a web browser to validate its deployment.

Part 3: Deploying to Linux behind a Load Balancer

- **Task**: Deploy the application on a Linux Server instance on AWS EC2 and configure it behind an Elastic Load Balancer.
- Create one or more Linux EC2 instances.
  - Do it manually for once, FINALLY, write the configuration to create all the AWS Resources using IaC (of your choice)
- Deploy the application to the Linux server(s), ensuring it runs as expected.
- Create an Elastic Load Balancer and configure it to distribute traffic to the Linux instance(s).

        - Do it manually for once, FINALLY, write the configuration to create all the AWS Resources using IaC (of your choice)
    - Test the load balancer's distribution by accessing the application through the load balancer's DNS name.

Food for thought
Can you automate the application deployment process, if yes how? What tools would you use and what would be the process?

4. Evaluation Criteria

- **Code Quality**: Is the application code well-structured, readable, and efficient?
- **Deployment Accuracy**: Were the deployments on both Windows and Linux successful? Does the application work as expected in both environments?
- **Understanding of AWS**: Does the setup utilize AWS services correctly? Are the instances and load balancer configured properly?
- **Documentation**: Is the process well-documented, including any setup steps, dependencies, and instructions to run the application?
- **Automation**: What are the levels of automation achieved while writing and creating the infrastructure and the deployment as well.

5. Submission Guidelines

- The answerer should provide a Git repository link containing the application source code, along with a README.md file containing:
- Instructions on how to run the application locally.
- Deployment instructions for both Windows and Linux environments.
- A brief explanation of the architecture and any design decisions.

- **An explanation of limitations of the architecture is needed and must be known.**

This test combines coding, deployment, and AWS cloud skills, offering a comprehensive assessment of the intern's capabilities. It's important to provide clear instructions and expectations while also leaving room for the intern to demonstrate problem-solving skills and creativity.

Guides:

1. How to create Free Tier Account on AWS : (do not forget to remove resources when you are done with your experiences) How to Create AWS Free Tier Account 2024
2. EC2 Instances in AWS : AWS How to Launch an EC2 instance? - Step by Step tutorial (Part-4)
3. What is VPC? : AWS VPC & Subnets | Amazon Web Services BASICS
4. What is VPC Routing : Amazon Virtual Private Cloud (VPC) Routing Deep Dive
5. What is Route Table?: AWS VPC Route Table
6. Again, VPC and Subnets for Beginners! AWS VPC & Subnets For Beginners
7. Recap 3 to 6 plus Private Subnet, NAT, Internet Gateway : AWS how to setup VPC, Public, Private Subnet, NAT, Internet Gateway, Route Ta…
8. Playlist: Terraform : What is Terraform? | Terraform Tutorial | #1
9. AWS Cloudformation : AWS CloudFormation Template Tutorial