```
# AWS Lambda function for request routing
resource "aws_lambda_function" "request_router" {
  function_name = "requestRouter"
  handler       = "index.handler"  # make sure this handler aligns with your code's entry point
  role          = aws_iam_role.lambda_exec_role.arn
  runtime       = "nodejs14.x"  # or whatever runtime you're using

  s3_bucket     = "swarmslambda"
  s3_key        = "code.zip"

  environment {
    variables = {
                                                    COGVLM_ENDPOINT               =
"http://${kubernetes_service.cogvlm_service.status[0].load_balancer[0].ingress[0].hostname}",
                                                    QWENVL_ENDPOINT               =
"http://${kubernetes_service.qwenvl_service.status[0].load_balancer[0].ingress[0].hostname}",
    }
  }
}

# API Gateway to expose the Lambda Function
resource "aws_api_gateway_rest_api" "model_routing_api" {
  name        = "ModelRoutingAPI"
  description = "API Gateway to route model requests"
}
```

```
resource "aws_api_gateway_resource" "model_routing_resource" {
  rest_api_id = aws_api_gateway_rest_api.model_routing_api.id
  parent_id   = aws_api_gateway_rest_api.model_routing_api.root_resource_id
  path_part   = "{proxy+}" # Enable proxy resource to capture all sub-paths
}

resource "aws_api_gateway_method" "model_post_method" {
  rest_api_id   = aws_api_gateway_rest_api.model_routing_api.id
  resource_id   = aws_api_gateway_resource.model_routing_resource.id
  http_method   = "ANY" # Accept any HTTP method
  authorization = "NONE"
}

resource "aws_api_gateway_integration" "model_lambda_integration" {
  rest_api_id = aws_api_gateway_rest_api.model_routing_api.id
  resource_id = aws_api_gateway_resource.model_routing_resource.id
  http_method = aws_api_gateway_method.model_post_method.http_method

  integration_http_method = "POST"
  type                    = "AWS_PROXY"
  uri                     = aws_lambda_function.request_router.invoke_arn
}

resource "aws_api_gateway_deployment" "api_deployment" {
  depends_on = [
    aws_api_gateway_integration.model_lambda_integration,
```

```
    aws_api_gateway_method.model_post_method,
    aws_api_gateway_resource.model_routing_resource,
    null_resource.update_lambda_env # Ensure API deployment waits for Lambda env vars update

  ]

  rest_api_id = aws_api_gateway_rest_api.model_routing_api.id
  stage_name  = "prod"
}

resource "aws_lambda_permission" "api_lambda_permission" {
  statement_id  = "AllowAPIGatewayInvoke"
  action        = "lambda:InvokeFunction"
  function_name = aws_lambda_function.request_router.function_name
  principal     = "apigateway.amazonaws.com"
  source_arn    = "${aws_api_gateway_rest_api.model_routing_api.execution_arn}/*/*"
}

resource "aws_api_gateway_stage" "api_stage" {
  deployment_id = aws_api_gateway_deployment.api_deployment.id
  rest_api_id   = aws_api_gateway_rest_api.model_routing_api.id
  stage_name    = "v1"
}
```