

## 1:- Create EC2 Instance

EC2 > Instances > Launch an instance

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Del

Browse more AMIs

Amazon Machine Image (AMI)  
Amazon Linux 2023 kernel-6.1 AMI  
ami-0f7b02bb6a0e14062 (64-bit (x86), uefi-preferred) / ami-03c1e90ffc89c0ce (64-bit (Arm), uefi)  
Virtualization: hvm    ENA enabled: true    Root device type: ebs

Description  
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.  
  
Amazon Linux 2023 AMI 2023.9.20251014.0 x86\_64 HVM kernel-6.1

Architecture: 64-bit (x86\_64)    Boot mode: uefi-preferred    AMI ID: ami-0f7b02bb6a0e14062    Publish Date: 2025-10-09    Username: ec2-user

Summary  
Number of instances: 1  
Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more  
ami-0f7b02bb6a0e14062  
Virtual server type (instance type): t2.micro  
Firewall (security group): New security group  
Launch instance

EC2 > Instances > Launch an instance

Instance type  
t2.micro  
Family: t2    1 vCPU    1 GiB Memory    Current generation: true  
On-Demand Ubuntu Pro base pricing: 0.0115 USD per Hour  
On-Demand Windows base pricing: 0.0178 USD per Hour  
On-Demand RHEL base pricing: 0.0276 USD per Hour  
On-Demand SUSE base pricing: 0.0132 USD per Hour  
On-Demand Linux base pricing: 0.0132 USD per Hour  
Free tier eligible  
All generations  
Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login)  
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.  
Key pair name - required  
adish-MERN-backend  
Create new key pair

Summary  
Number of instances: 1  
Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more  
ami-0f7b02bb6a0e14062  
Virtual server type (instance type): t2.micro  
Firewall (security group): New security group  
Launch instance

EC2 > Instances > Launch an instance

Network settings  
Network: vpc-0376ebe6043cd8004  
Subnet: No preference (Default subnet in any availability zone)  
Auto-assign public IP: Enable  
Firewall (security groups)  
Create security group  
We'll create a new security group called 'launch-wizard-59' with the following rules:  
Allow SSH traffic from: Anywhere  
Allow HTTPS traffic from the internet  
Allow HTTP traffic from the internet  
Launch instance

Summary  
Number of instances: 1  
Software Image (AMI): Amazon Linux 2023 AMI 2023.9.2...read more  
ami-0f7b02bb6a0e14062  
Virtual server type (instance type): t2.micro  
Firewall (security group): New security group  
Launch instance

## 2:- Create ALB Load Balancer

The screenshot shows the 'Create Application Load Balancer' page in the AWS Management Console. The breadcrumb navigation is 'EC2 > Load balancers > Create Application Load Balancer'. The page is divided into sections for VPC, IP pools, and Availability Zones and subnets.

**VPC** | Info  
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#).

vpc-0376ebe6043cd8004 (default) Create VPC

**IP pools** | Info  
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view Pools in the [Amazon VPC IP Address Manager console](#).

☒ Use IPAM pool for public IPv4 addresses  
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

**Public IPv4 IPAM pool**  
Choose an IPAM pool  
The locale of the IPAM pool must be equal to the current Region and the awsService attribute must be EC2.

**Availability Zones and subnets** | Info  
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

☒ eu-west-2a (euw2-az2)  
Subnet  
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.

subnet-067a0303e6a0bb68f  
IPv4 subnet CIDR: 172.31.16.0/20

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the 'Load balancers' page in the AWS Management Console. The breadcrumb navigation is 'EC2 > Load balancers'. The page displays a table of load balancers and a detailed view of the 'ELB5xxErrors' load balancer.

**Load balancers (1/1)** Actions Create load balancer

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

<input checked="" type="checkbox"/>	Name	State	Type	Scheme	IP address type	VPC ID
<input checked="" type="checkbox"/>	ELB5xxErrors	Active	application	Internet-facing	IPv4	vpc-0376ebe6043cd8000

**Load balancer: ELB5xxErrors**

Details Listeners and rules Network mapping Resource map Security Monitoring Integrations

**Details**

Load balancer type	Status	VPC	Load balancer IP address type
Application	Active	vpc-0376ebe6043cd8000	IPv4

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 3 :- Create SNS

The screenshot shows the 'adish-invoice-topic' page in the AWS Management Console. The breadcrumb navigation is 'Amazon SNS > Topics > adish-invoice-topic'. The page displays details about the topic and a list of subscriptions.

**adish-invoice-topic** Edit Delete Publish message

**Details**

Name	Display name
adish-invoice-topic	-

**ARN**  
arn:aws:sns:eu-west-2:975050024946:adish-invoice-topic

**Topic owner**  
975050024946

**Type**  
Standard

**Subscriptions (2)** Edit Delete Request confirmation Confirm subscription Create subscription

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 4:- Create IAM Role

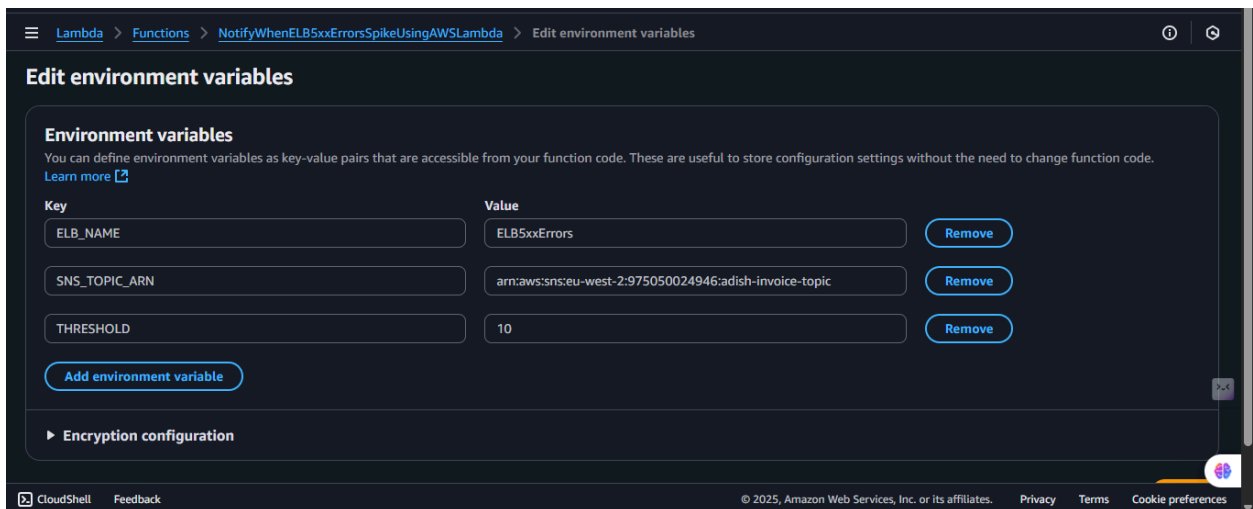
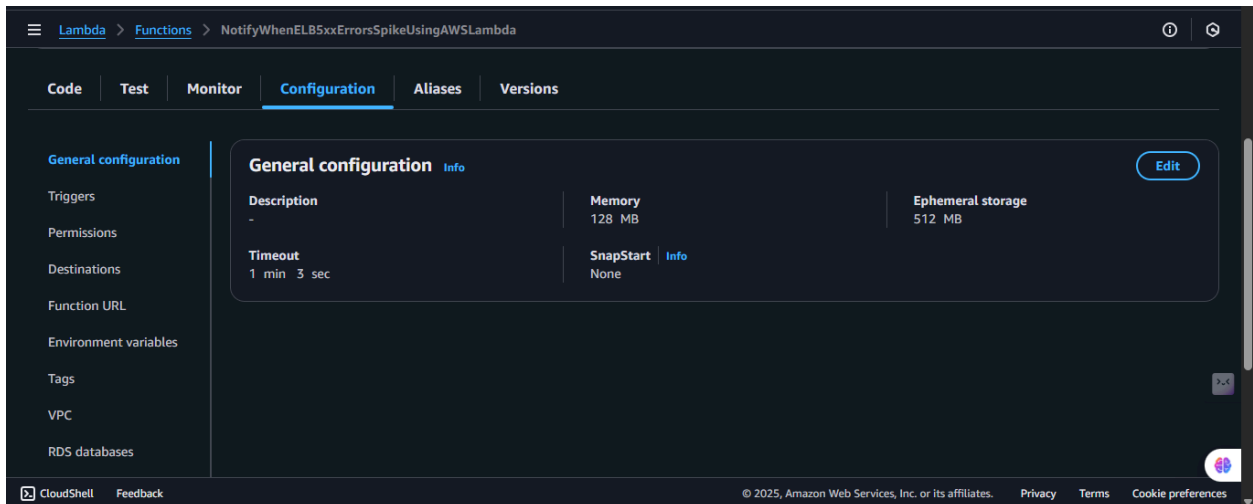
The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar with 'Identity and Access Management (IAM)' selected. The main content area is titled 'Permissions policies (7)' and includes a search bar, a 'Filter by Type' dropdown set to 'All types', and a table of policies. The table lists seven AWS managed policies with their names, types, and the number of attached entities.

Policy name	Type	Attached entities
AmazonCloudWatchEvidentlyR...	AWS managed	2
AmazonEC2ContainerRegistry...	AWS managed	25
AmazonEC2FullAccess	AWS managed	115
AmazonS3ObjectLambdaExec...	AWS managed	1
AmazonSNSFullAccess	AWS managed	34
AWSLambdaBasicExecutionRole	AWS managed	68

## 5:- Create Lambda Function

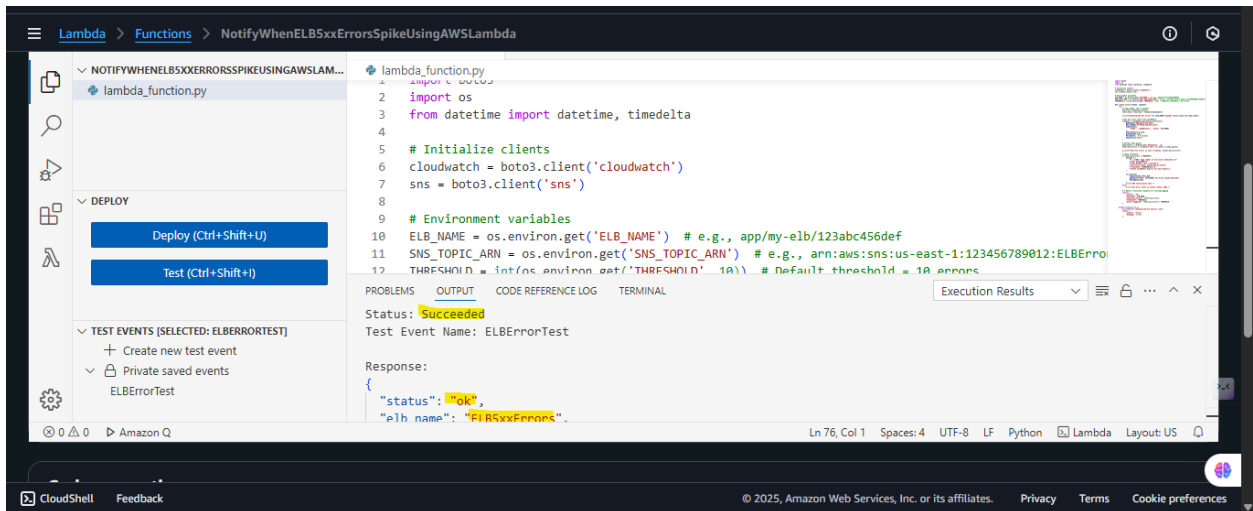
The screenshot shows the 'Create function' page in the AWS Lambda console. The 'Author from scratch' option is selected. The 'Basic information' section is expanded, showing the function name 'NotifyWhenELB5xxErrorsSpikeUsingAWSLambda', the runtime 'Python 3.13', and the architecture 'x86\_64'.

The screenshot shows the 'Permissions' tab of the 'Create function' page. It explains that Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. The 'Change default execution role' section is expanded, showing the 'Execution role' options: 'Create a new role with basic Lambda permissions', 'Use an existing role' (selected), and 'Create a new role from AWS policy templates'. The 'Existing role' dropdown is set to 'Jigar-Role-ELB-AutoScale-Notifications'.



# Environment Variables

Variable Name	Example Value	Description
ELB_NAME	app/my-elb/123abc456def	Your Load Balancer name (from AWS console)
SNS_TOPIC_ARN	arn:aws:sns:us-east-1:123456789012:ELBErrorAlerts	SNS Topic ARN
THRESHOLD	10	5xx error threshold for triggering alert



```
import boto3
import os
from datetime import datetime, timedelta

# Initialize clients
cloudwatch = boto3.client('cloudwatch')
sns = boto3.client('sns')

# Environment variables
ELB_NAME = os.environ.get('ELB_NAME') # e.g., app/my-elb/123abc456def
SNS_TOPIC_ARN = os.environ.get('SNS_TOPIC_ARN') # e.g.,
arn:aws:sns:us-east-1:123456789012:ELBErrorAlerts
THRESHOLD = int(os.environ.get('THRESHOLD', 10)) # Default threshold = 10
errors

def lambda_handler(event, context):
    try:
        # Time window: last 5 minutes
        end_time = datetime.utcnow()
        start_time = end_time - timedelta(minutes=5)

        print(f"Checking ELB 5xx errors for {ELB_NAME} between
{start_time} and {end_time}")

        # Get 5xx error count from CloudWatch
        response = cloudwatch.get_metric_statistics(
            Namespace='AWS/ApplicationELB',
```

```

MetricName='HTTPCode_ELB_5XX_Count',
Dimensions=[
    {'Name': 'LoadBalancer', 'Value': ELB_NAME}
],
StartTime=start_time,
EndTime=end_time,
Period=300, # 5 minutes
Statistics=['Sum']
)

# Extract data points
data_points = response.get('Datapoints', [])
total_5xx_errors = sum(point['Sum'] for point in data_points)

print(f"Total 5xx errors in last 5 minutes: {total_5xx_errors}")

# Check threshold
if total_5xx_errors > THRESHOLD:
    message = (
        f"⚠️ ALERT: High number of 5xx errors detected!\n\n"
        f"ELB: {ELB_NAME}\n"
        f"Time Window: Last 5 minutes\n"
        f"5xx Error Count: {total_5xx_errors}\n"
        f"Threshold: {THRESHOLD}\n\n"
        f"Check CloudWatch metrics for more details."
    )

    sns.publish(
        TopicArn=SNS_TOPIC_ARN,
        Subject=f"ALERT: {ELB_NAME} 5xx Errors Spike Detected",
        Message=message
    )
    print("SNS notification sent.")
else:
    print("5xx error count is within normal range.")

# ✅ Return structured response for testing/logging
return {
    "status": "ok",
    "elb_name": ELB_NAME,

```

```

        "total_5xx_errors": total_5xx_errors,
        "threshold": THRESHOLD,
        "alert_triggered": total_5xx_errors > THRESHOLD
    }

except Exception as e:
    print(f"Error checking ELB 5xx metrics: {e}")
    return {
        "status": "error",
        "message": str(e)
    }

```

