```
import boto3, os, sys, json, logging

# Set the global variables
globalVars  = {}

globalVars['REGION_NAME']           = "us-east-1"
globalVars['SNSTopicArn']           = "ENTER YOUR SNS ARN"

sns_client = boto3.client('sns')

# Set the log format
logger = logging.getLogger()
for h in logger.handlers:
  logger.removeHandler(h)

h = logging.StreamHandler(sys.stdout)
FORMAT = ' [%(levelname)s]/%(asctime)s/%(name)s - %(message)s'
h.setFormatter(logging.Formatter(FORMAT))
logger.addHandler(h)
logger.setLevel(logging.INFO)

"""
If User provides different values, override defaults
"""
def setGlobalVars():
    try:
        if os.environ['SNSTopicArn']:
            globalVars['SNSTopicArn']  = os.environ['SNSTopicArn']
    except KeyError as e:
        logger.error('ERROR: SNS Topic ARN is missing, Using default GlobalVars - {0}'.format(
globalVars['SNSTopicArn'] ) )
        logger.error('ERROR: {0}'.format( str(e) ) )
        pass

"""
This function pushes GuardDuty *Findings* to SNS Topic to be picked up ITSM Tools for
Alerting.
"""

def push_To_SNS_Topic(event):
    try:
        response = sns_client.publish(
        TopicArn = globalVars['SNSTopicArn'],
        Message = json.dumps(event),
        Subject = event['detail']['title']
        )
        logger.info('SUCCESS: Pushed GuardDuty Finding to SNS Topic')
        return "Successly pushed to Notification to SNS Topic"
    except KeyError as e:
        logger.error('ERROR: Unable to push to SNS Topic: Check [1] SNS Topic ARN is invalid, [2]
IAM Role Permissions{0}'.format( str(e) ) )
        logger.error('ERROR: {0}'.format( str(e) ) )


def lambda_handler(event, context):
    setGlobalVars()
    return push_To_SNS_Topic(event)

if __name__ == '__main__':
    lambda_handler(None, None)
```