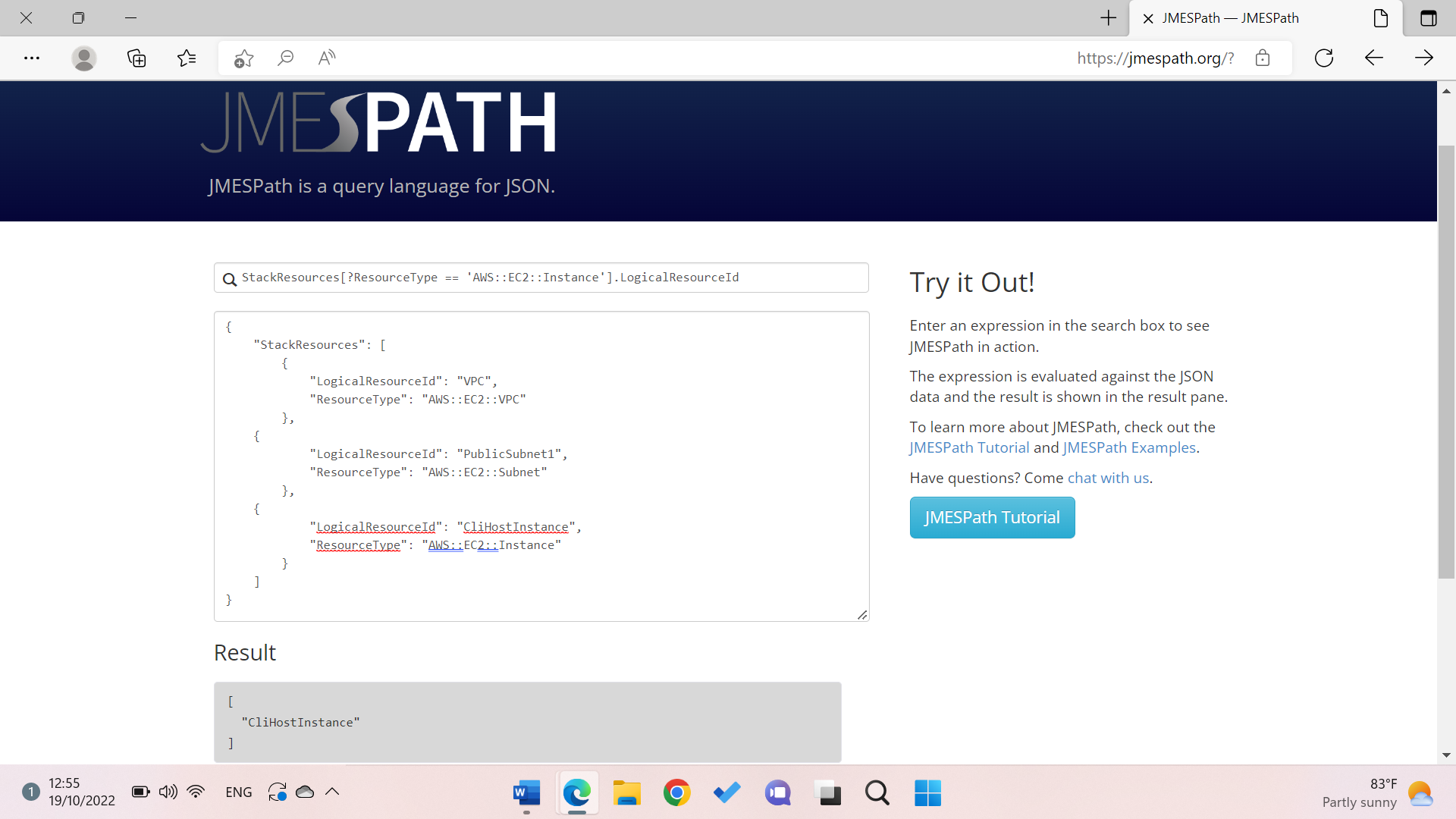
Troubleshoot CloudFormation

**Task 1: Practice querying JSON-formatted data by using JMESPath**



**Task 2: Troubleshooting and working with AWS CloudFormation stacks**

establish an SSH connection to the CLI Host to work with the AWS CloudFormation service and configure the AWS CLI.

Then create a stack by using this command:

aws cloudformation create-stack \

--stack-name myStack \

--template-body file://template1.yaml \

--capabilities CAPABILITY\_NAMED\_IAM \

--parameters ParameterKey=KeyName,ParameterValue=vockey

And check if the stack is created:

watch -n 5 -d \

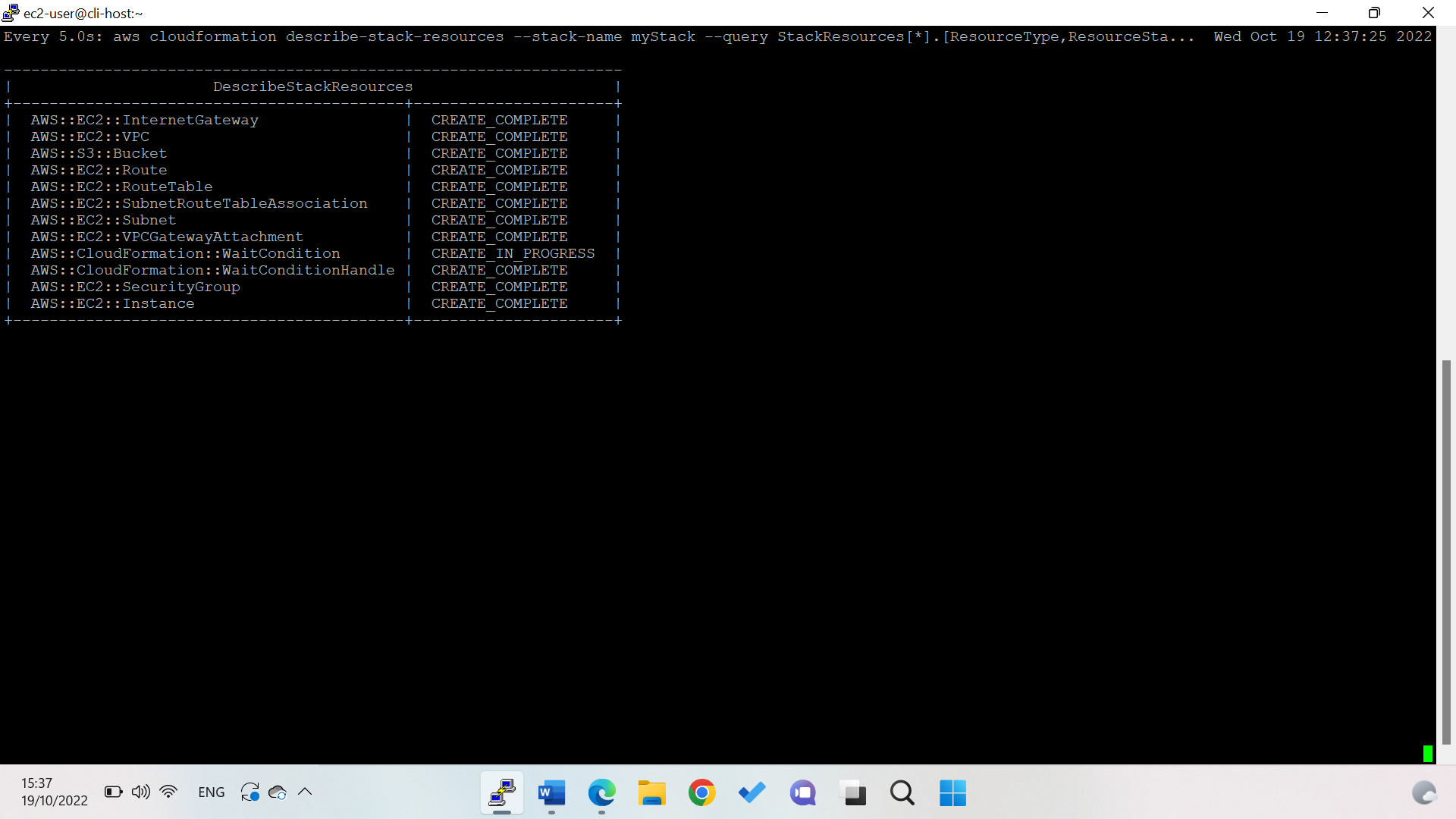
aws cloudformation describe-stack-resources \

--stack-name myStack \

--query 'StackResources[\*].[ResourceType,ResourceStatus]' \

--output table

**Output:**



A screenshot of a computer

Description automatically generated

Notice: that after almost all resources are created, they start being deleted.

To see the stack status and other details, run the describe-stacks command:

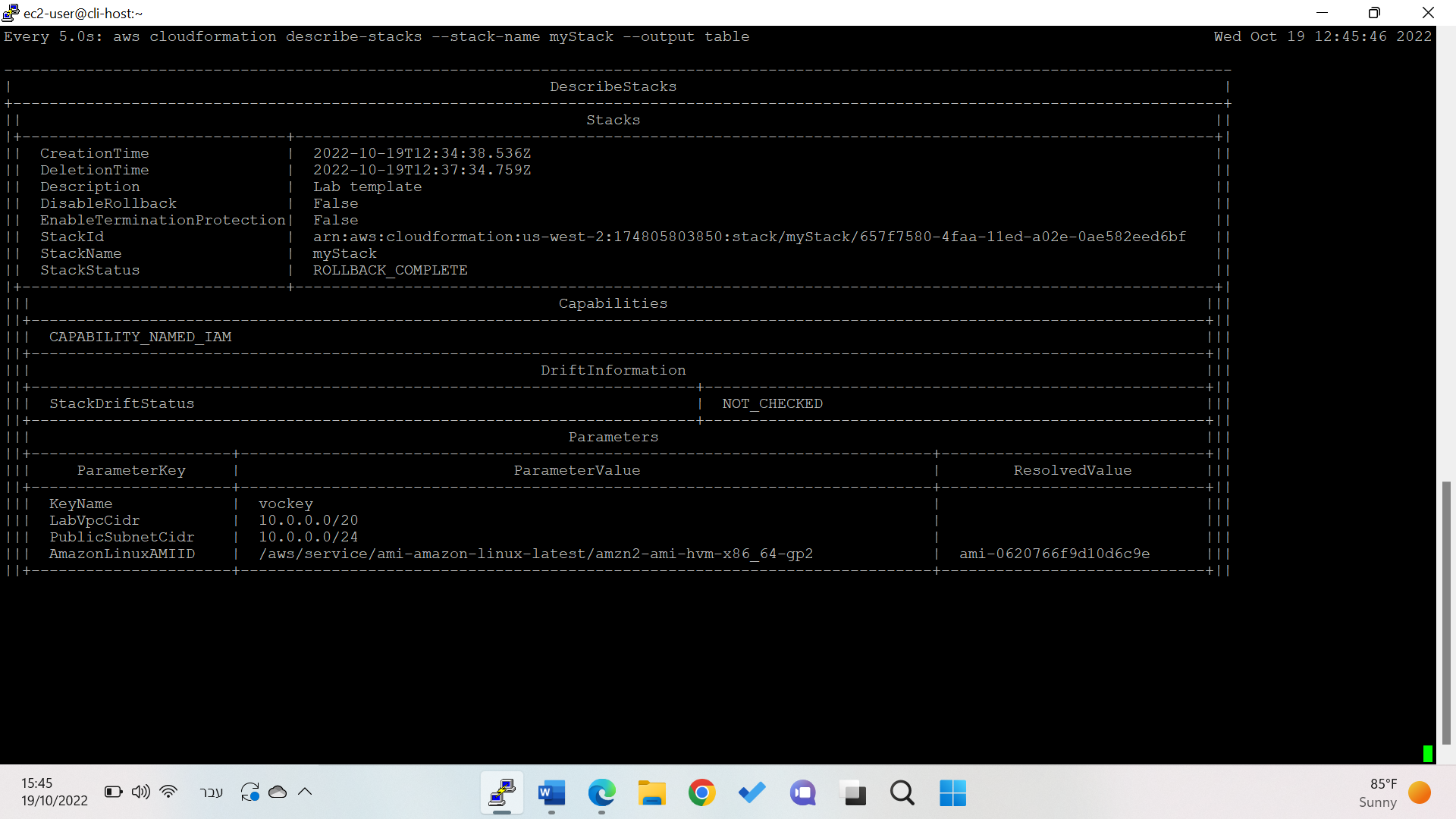
watch -n 5 -d \

aws cloudformation describe-stacks \

--stack-name myStack \

--output table

Output:



Analyze the issue by running the describe-stack-events command with a query that returns only the CREATE\_FAILED events:

aws cloudformation describe-stack-events \

--stack-name myStack \

--query "StackEvents[?ResourceStatus == 'CREATE\_FAILED']"

Output: timeout- This situation could indicate that there is an issue in the **userdata** section of the EC2 instance resource that the template attempted to create.

Text

Description automatically generated

Run:

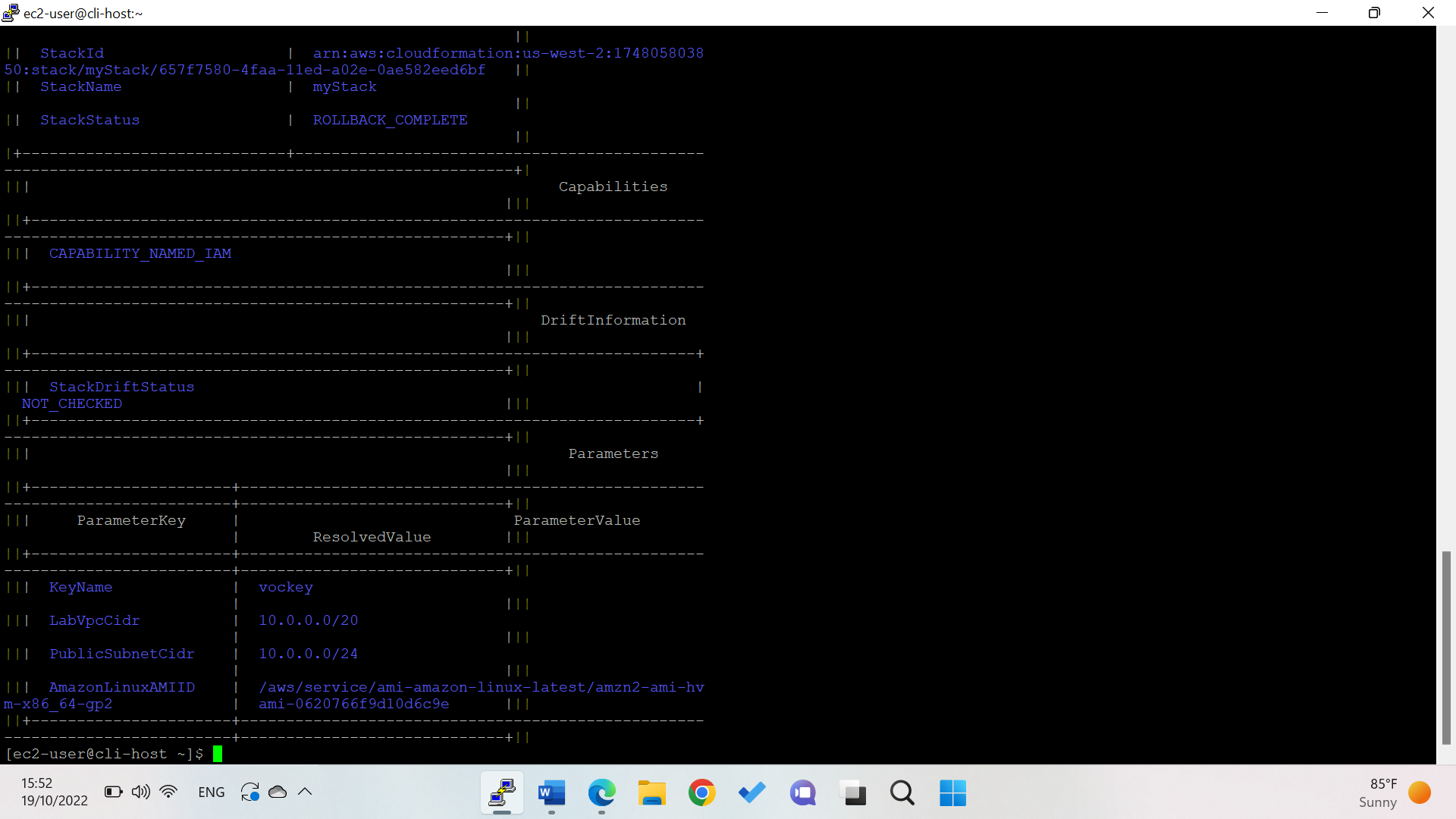
aws cloudformation describe-stacks \

--stack-name myStack \

--output table

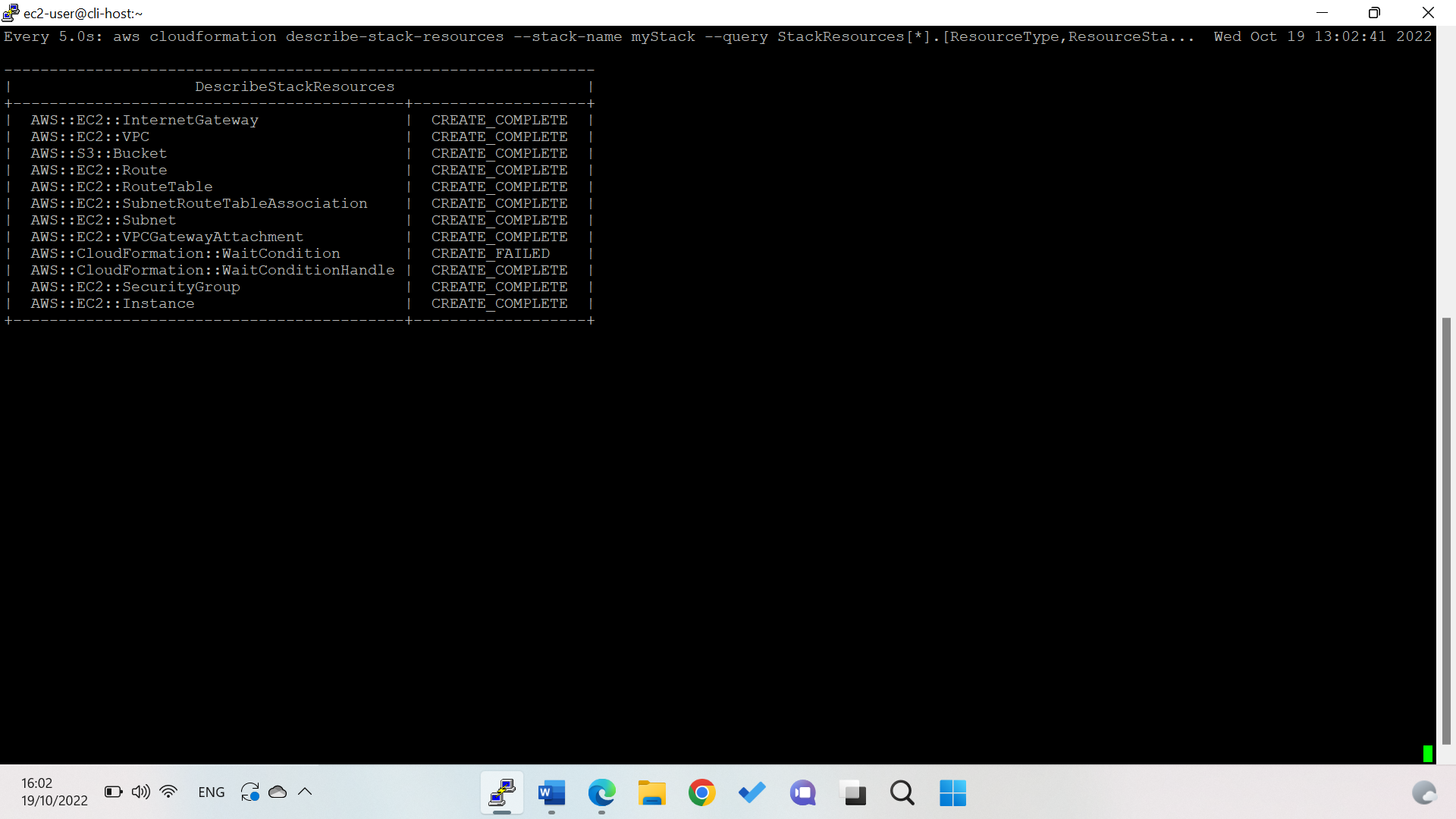
Output: The StackStatus element indicates that the status of the stack is now *ROLLBACK\_COMPLETE*.

This status confirms that the EC2 instance and the other resources that were created by the stack were deleted.



**Task 2.4: Avoid rollback on an AWS CloudFormation stack**

When creating a stack specify: --on-failure DO\_NOTHING. This parameter will ensure that a rollback will not occur this time. Then Run the describe-stack-resources command again and notice:



Now connect to the web serves instance by the IP using this command:

aws ec2 describe-instances \

--filters "Name=tag:Name,Values='Web Server'" \

--query 'Reservations[].Instances[].[State.Name,PublicIpAddress]'

Analyze the cloud-init-output.log file by running: sudo tail -50 /var/log/cloud-init-output.log





Run the following command to view the part-001 script: sudo cat /var/lib/cloud/instance/scripts/part-001

Output:

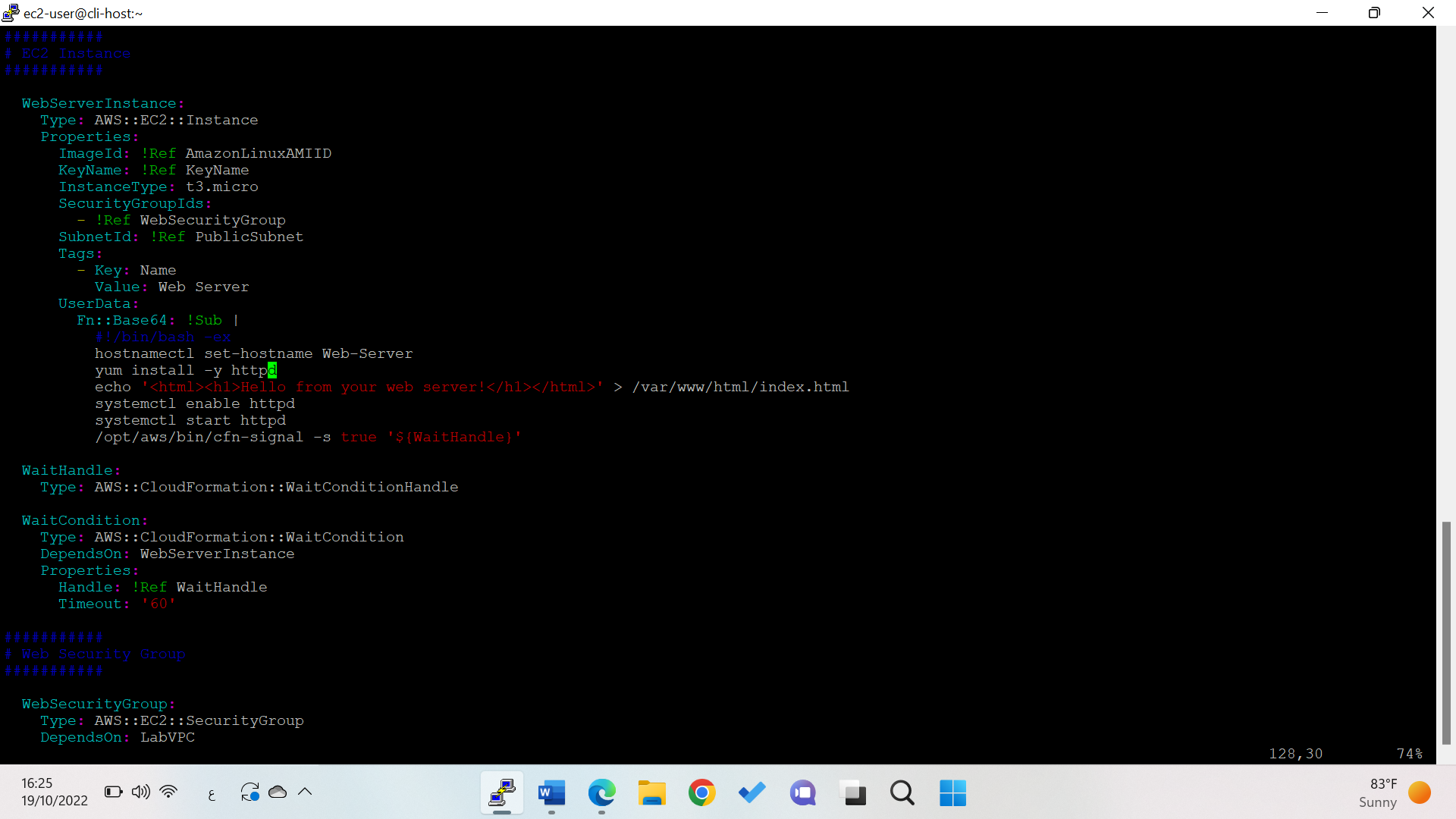
A screenshot of a computer

Description automatically generated

* In particular, notice the first line, which starts with the #! characters. This line includes the -e parameter. This parameter signals that if any command in the script fails, the whole script should immediately stop running with a non-zero status.
* In summary, because no package named *http* could be found, the userdata script failed. Therefore, the wait condition never received the success signal, and after 2 minutes, the wait condition timed out. This reason is why the stack failed.

### Task 2.5: Fix the issue and successfully create the AWS CloudFormation stack

update the AWS CloudFormation template: vim template1.yaml



Change http to httpd

Then confirm the change by running: cat template1.yaml | grep httpd

Output:

Graphical user interface, text

Description automatically generated

Delete the stack: aws cloudformation delete-stack --stack-name myStack

Then describe the stack:

watch -n 5 -d \

aws cloudformation describe-stacks \

--stack-name myStack \

--output table

Run the stack again:

aws cloudformation create-stack \

--stack-name myStack \

--template-body file://template1.yaml \

--capabilities CAPABILITY\_NAMED\_IAM \

--on-failure DO\_NOTHING \

--parameters ParameterKey=KeyName,ParameterValue=vockey

This time, your stack should be created successfully (without errors), and it should have a StackStatus of CREATE\_COMPLETE.

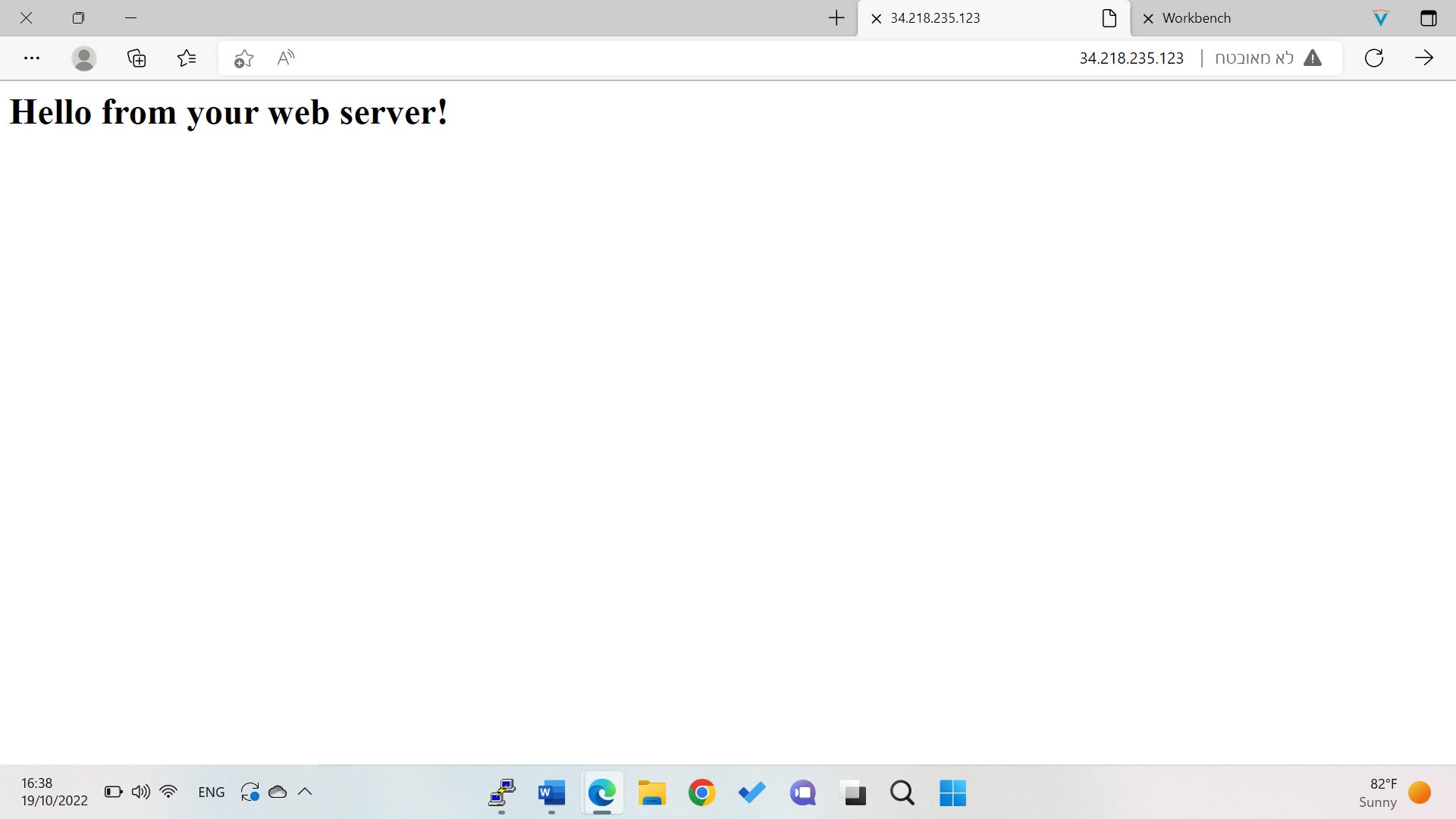
A screenshot of a computer

Description automatically generated

Run the describe-stacks command: and notice **Outputs** section includes the PublicIP address of the web server and the name of the S3 bucket that was created.

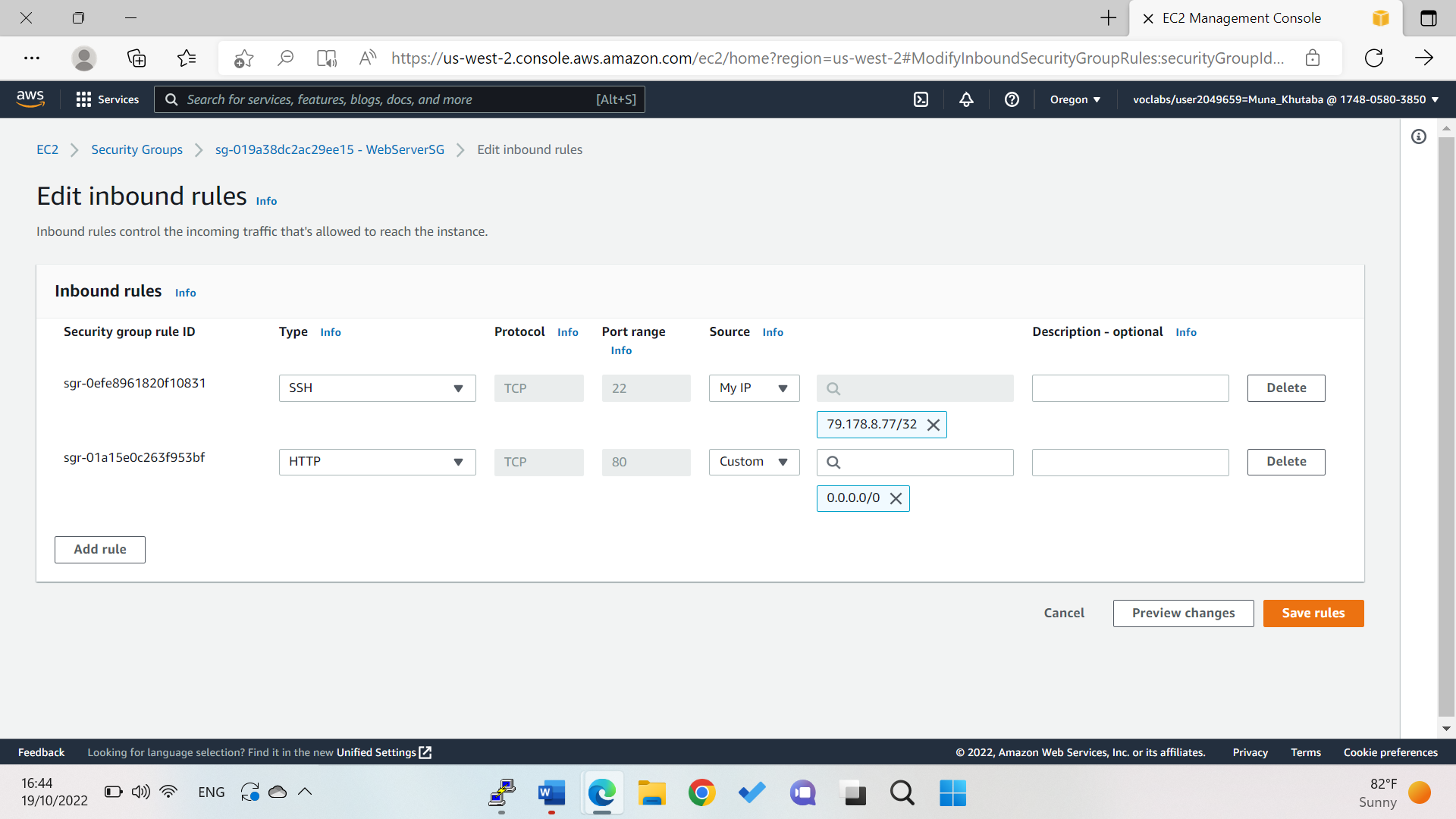
Text

Description automatically generated

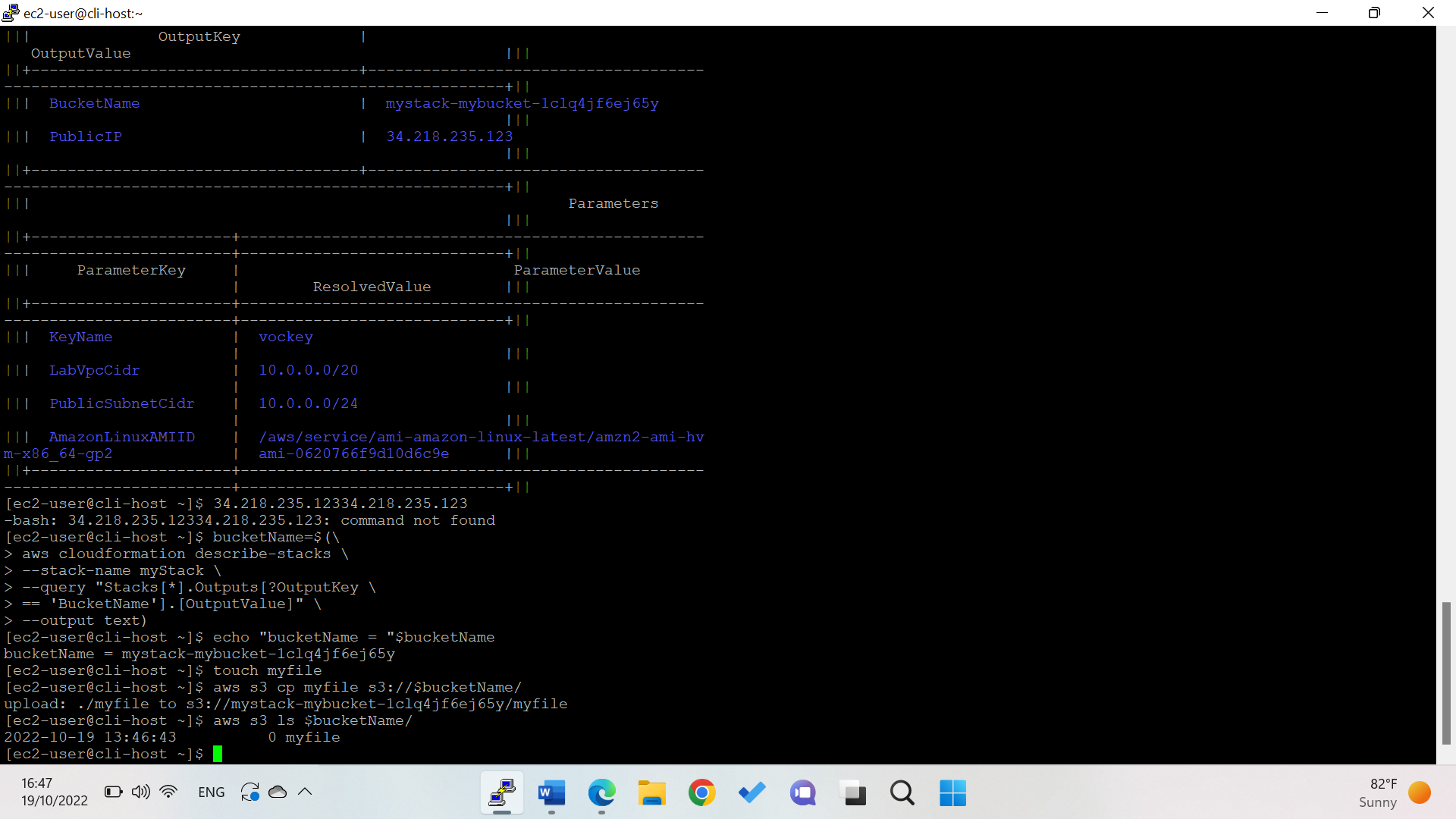
Copy the public ip address to a new browser: 34.218.235.123

## Task 3: Make manual modifications and detect drift

### Task 3.1: Make manual modifications to the security groups

Go to the web server ec2 instance, then security group and edit the inbound rule to 

### Task 3.2: Add an object to the S3 bucket.



connected to the CLI Host, query the bucket name, assign it to a variable named *bucketName*, then create an empty file, and copy the file to the bucket which uses the *bucketName* variable, finally, verify that the file is in the bucket.

### Task 3.3: Detect drift

To start drift detection on your stack, run the following command:

aws cloudformation detect-stack-drift --stack-name myStack

Output:

A screenshot of a computer

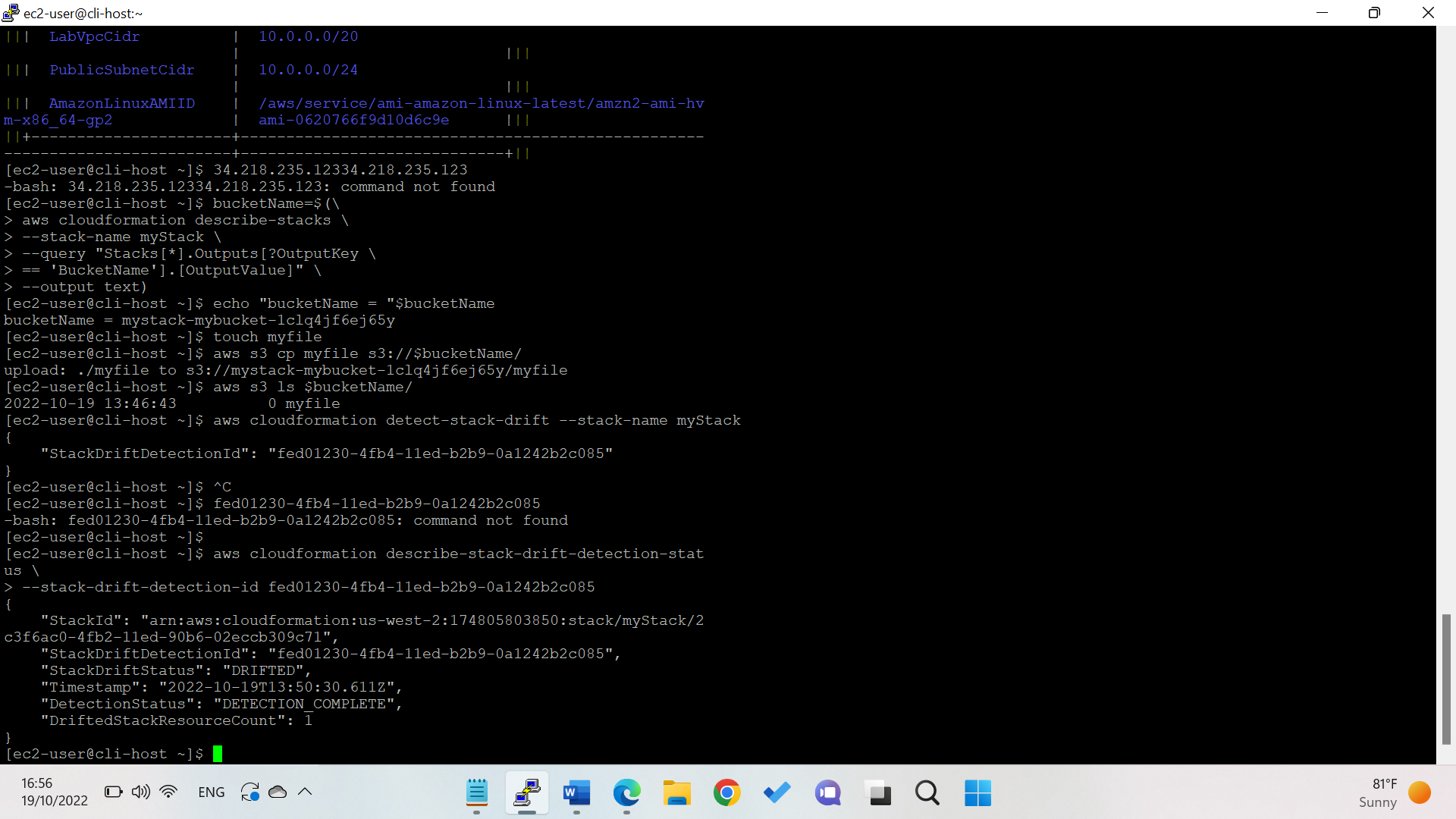
Description automatically generated with medium confidence

Monitor the status of drift by:

aws cloudformation describe-stack-drift-detection-status \

--stack-drift-detection-id fed01230-4fb4-11ed-b2b9-0a1242b2c085

**Output:**



Run the following commandthat will return only the resource type, resource status, and drift status:

aws cloudformation describe-stack-resources \

--stack-name myStack \

--query 'StackResources[\*].[ResourceType,ResourceStatus,DriftInformation.StackResourceDriftStatus]' \

--output table

A screenshot of a computer

Description automatically generated

Retrieve the specific details of the drift for the resource that has a StackResourceDriftStatus of MODIFIED:

aws cloudformation describe-stack-resource-drifts \

--stack-name myStack \

--stack-resource-drift-status-filters MODIFIED

Output:

Text

Description automatically generated

## Task 4: Attempt to delete the stack

Delet the stack then observe the results by the two commands:

aws cloudformation delete-stack --stack-name myStack

watch -n 5 -d \

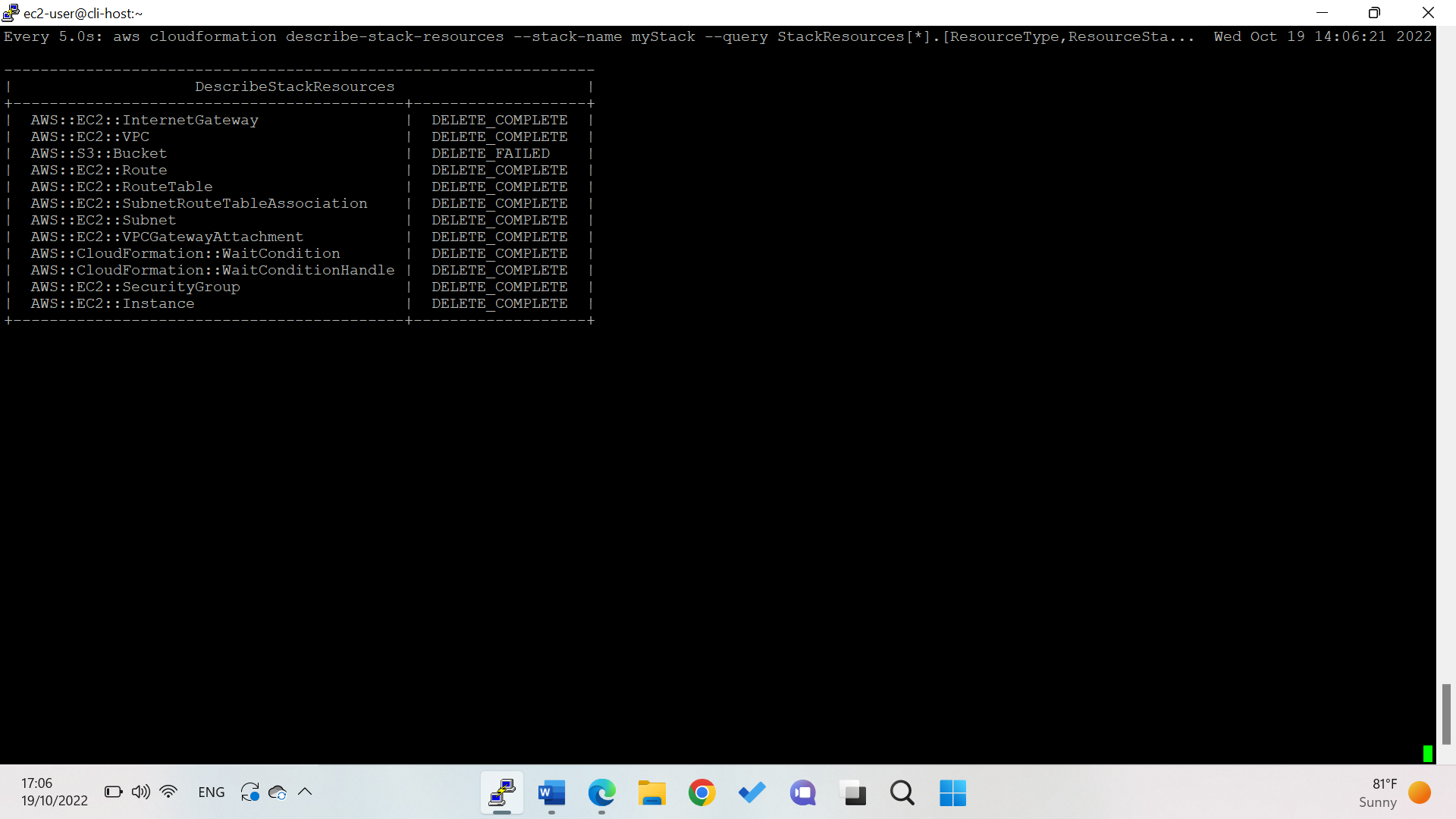
aws cloudformation describe-stack-resources \

--stack-name myStack \

--query 'StackResources[\*].[ResourceType,ResourceStatus]' \

--output table

Output: notice that the s3 bucket is deleted\_failed

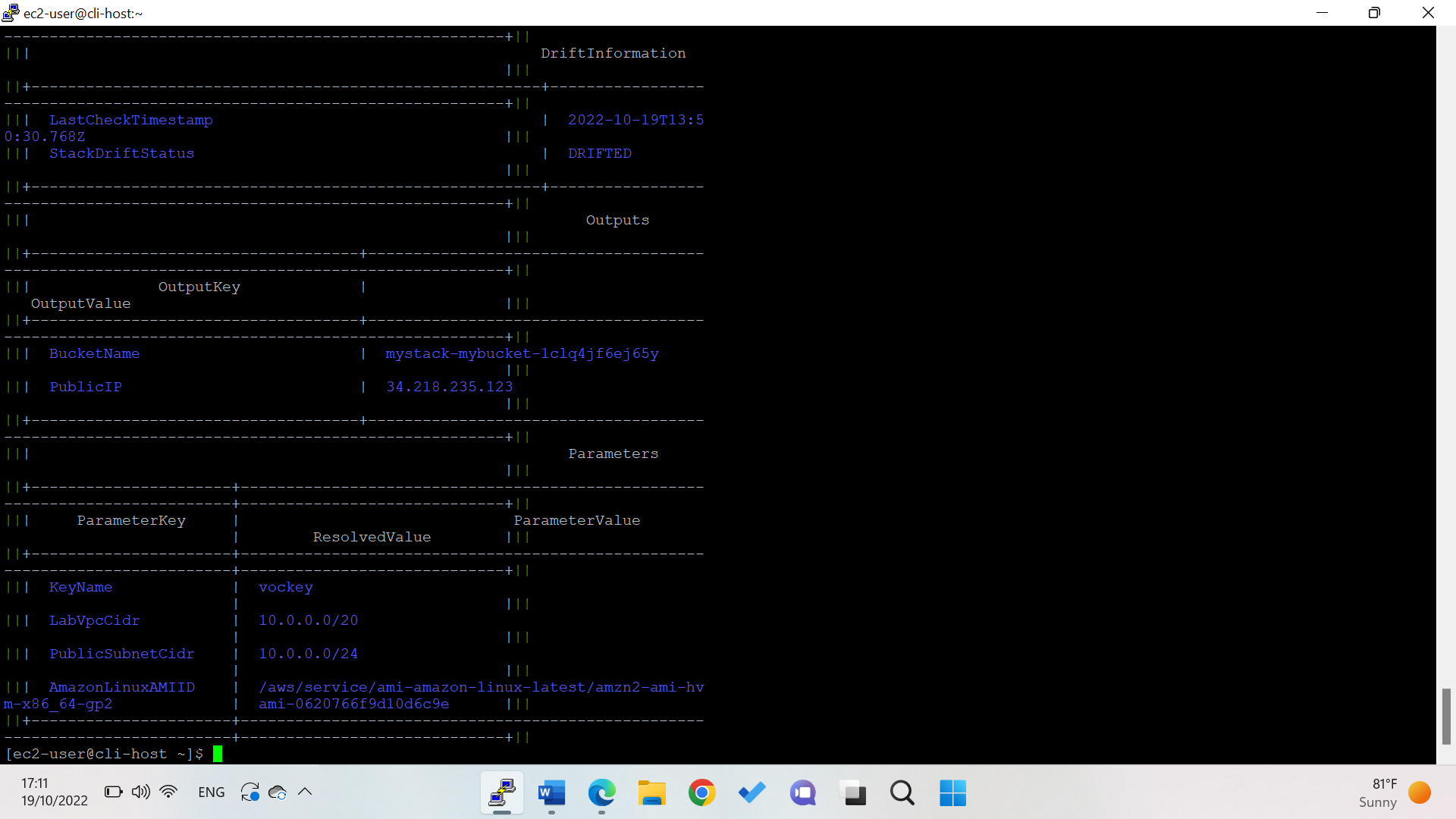


Then describe-stacks command to see the stack status:

aws cloudformation describe-stacks \

--stack-name myStack \

--output table

Output: 

CloudFormation will not delete a bucket that has objects in it. This is to help guard against accidental data loss.