
AWS
re:Invent

SEC317

Ten easy and effective ways to secure your AWS environment

Becky Weiss
Senior Principal Engineer
Amazon Web Services

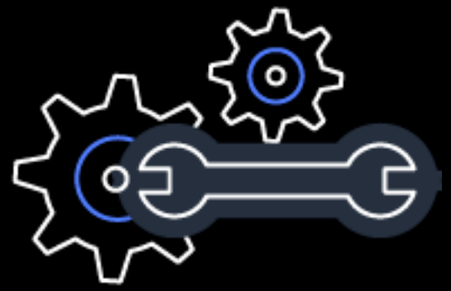
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You're in the right place if you

- Have some background in basic AWS and security concepts
- Want to talk builder to builder about **practical, concrete** steps you can take
- … And want to get through it quick!

Our focus: preventive security AWS skills



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our ten speed topics

1. Amazon S3 Block Public Access
2. Use federation – make all your IAM credentials temporary
3. Collect AWS CloudTrail for your entire AWS organization
4. Know how to query a CloudTrail
5. Tag your subnets for scalable control over connectivity
6. Centrally manage network security with AWS Firewall Manager
7. Assert network origin in your AWS Identity and Access Management (IAM) policies
8. Use the network to keep your data where you want
9. Connect to your Amazon Elastic Compute Cloud (Amazon EC2) instances without SSH keys
10. Monitor DNS from your VPC with Amazon Route 53 Resolver Query Logging

#1 Amazon S3 Block Public Access

IN ALL BUT A FEW, SPECIFIC CASES, YOUR DATA ISN'T PUBLIC. HAVE AWS REQUIRE PRIVATE ACCESS.



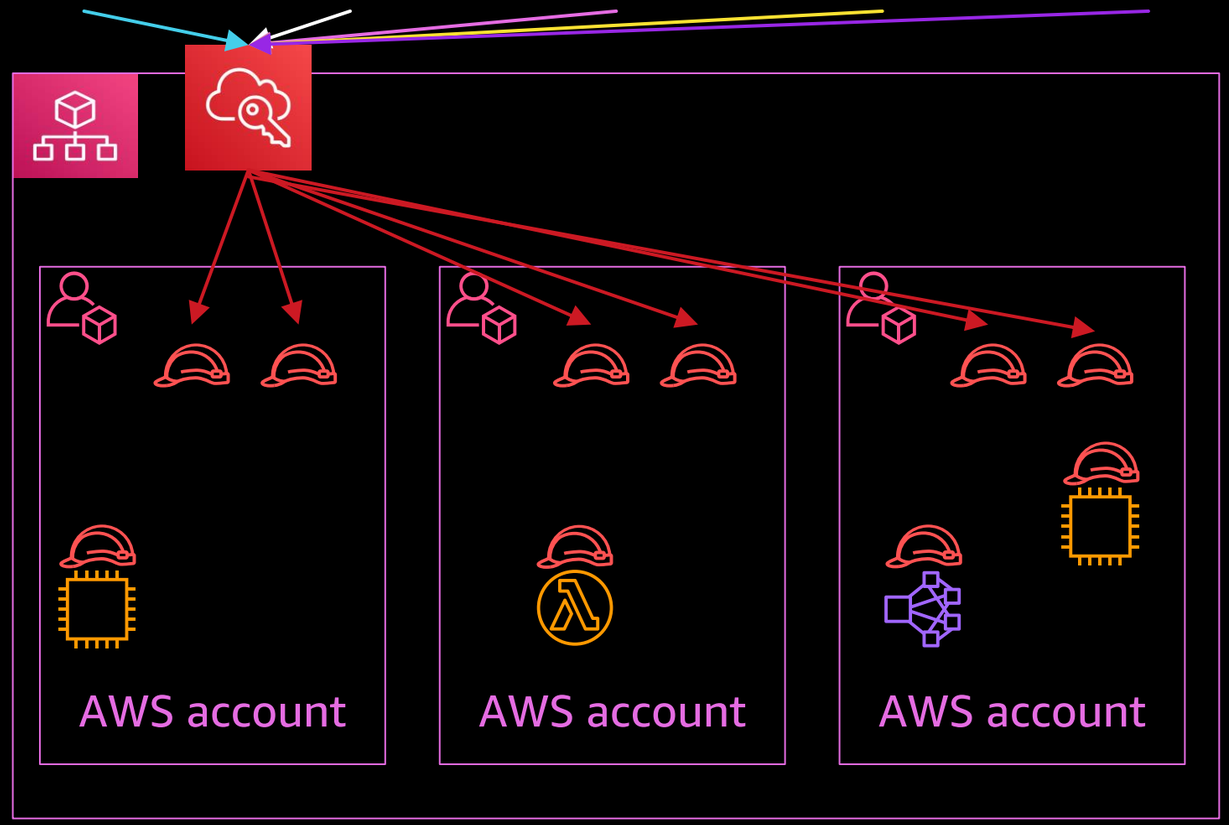
There are four settings.
Enable all of them.

#2 Make all your IAM credentials temporary

IAM ROLES WORK FOR BOTH HUMAN AND APPLICATION ACCESS

Human access:

- AWS Single Sign-On user pools and integrated identity providers – includes AWS Command Line Interface access
- Direct federation from identity providers such as Active Directory Federation Services

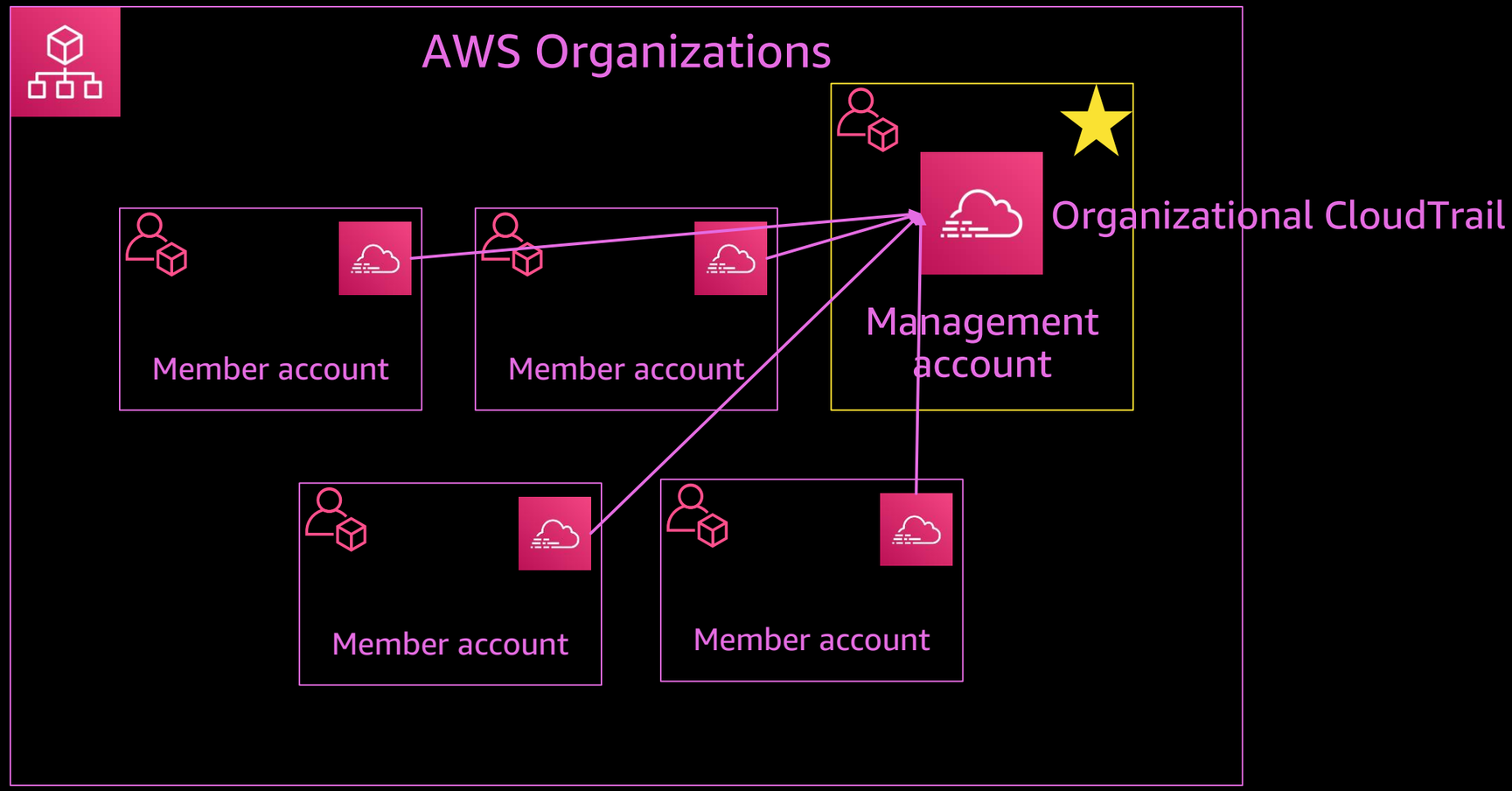


Application access:

Compute environments in AWS will obtain short-term credentials on your behalf and make them available for programmatic access

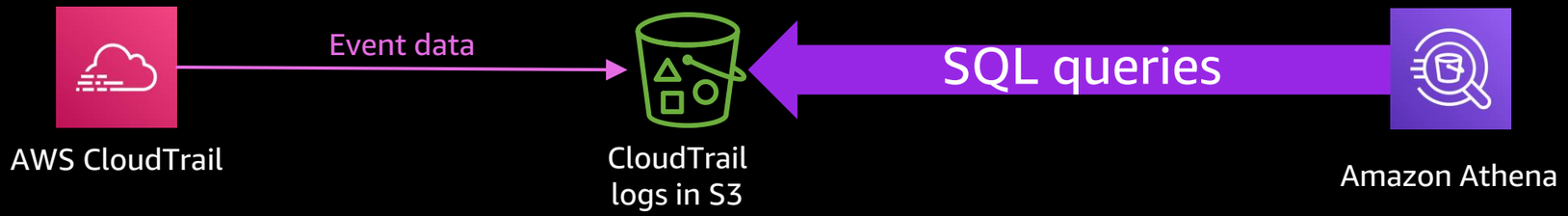
#3 Configure an Organizations-wide CloudTrail

ENABLE AUDIT LOGS EVERYWHERE AND CONSOLIDATE THEM IN ONE PLACE



#4 Know how to query CloudTrail with AWS Athena

THIS SKILL WILL HELP YOU WRITE GOOD IAM POLICIES AND ANSWER QUESTIONS ABOUT YOUR PATTERNS

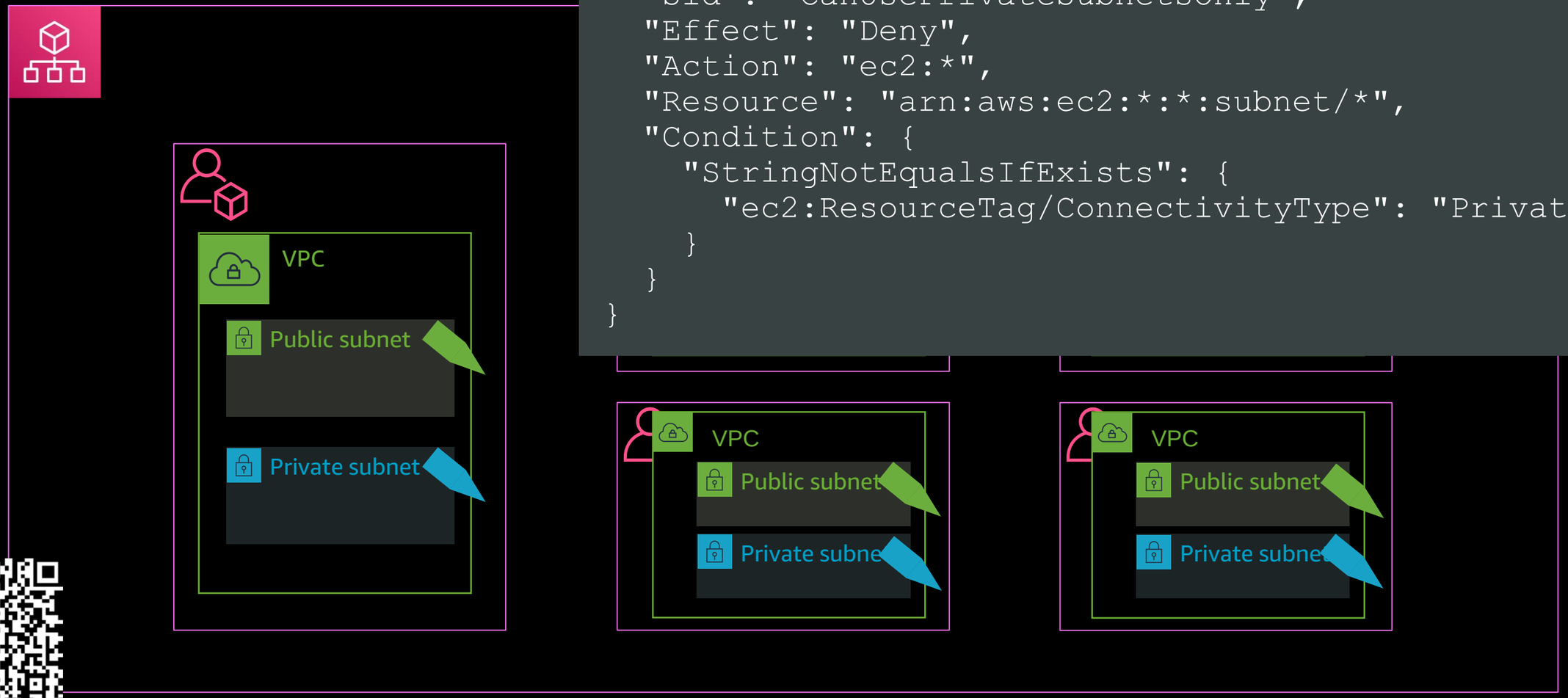



```
# What actions were taken under the "MyRole" identity on Aug 1?
➢ SELECT eventsource, eventname, COUNT(*) AS count FROM "default"."cloudtrail_logs"
   WHERE useridentity.sessioncontext.sessionissuer.arn = 'arn:aws:iam::111122223333:role/MyRole'
   AND eventtime > '2020-08-01T00:00:00Z' AND eventtime < '2020-08-02T00:00:00Z'
   GROUP BY eventname, eventsource
   ORDER BY count DESC;

# Which encryption keys did this person use to read data?
➢ SELECT eventtime, resources FROM "default"."cloudtrail_logs"
   WHERE eventsource = 'kms.amazonaws.com'
   AND eventname = 'Decrypt'
   AND useridentity.principalid = 'AROAJEXAMPLE:fred'
   AND eventtime > '2020-08-01T00:00:00Z' AND eventtime < '2020-08-01T01:00:00Z'
   ORDER BY eventtime;
```

#5 Tag your subnets according to connectivity

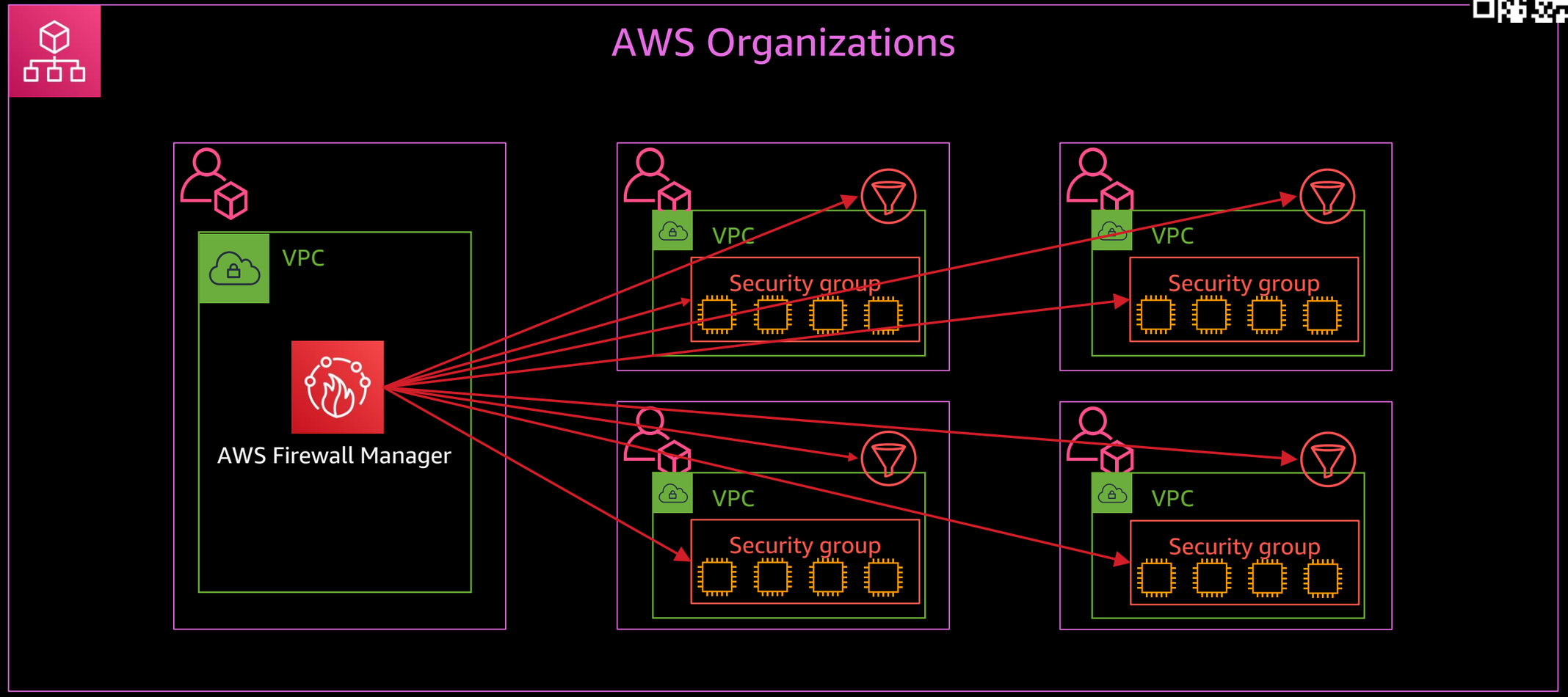
YOU'LL GET SCALABLE IAM CONTROLS OVER CONNECTIVITY




```
{
  "Sid": "CanUsePrivateSubnetsOnly",
  "Effect": "Deny",
  "Action": "ec2:*",
  "Resource": "arn:aws:ec2:*:*:subnet/*",
  "Condition": {
    "StringNotEqualsIfExists": {
      "ec2:ResourceTag/ConnectivityType": "Private"
    }
  }
}
```

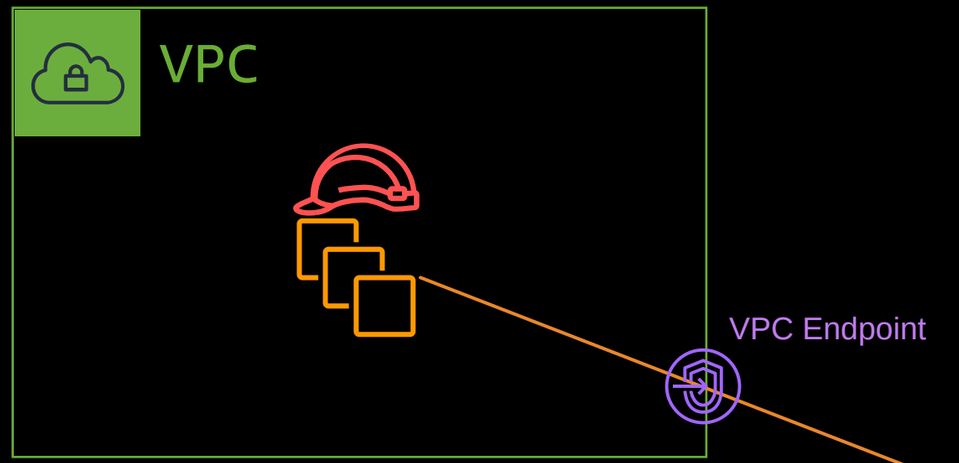
#6 Centralize network controls with AWS Firewall Manager

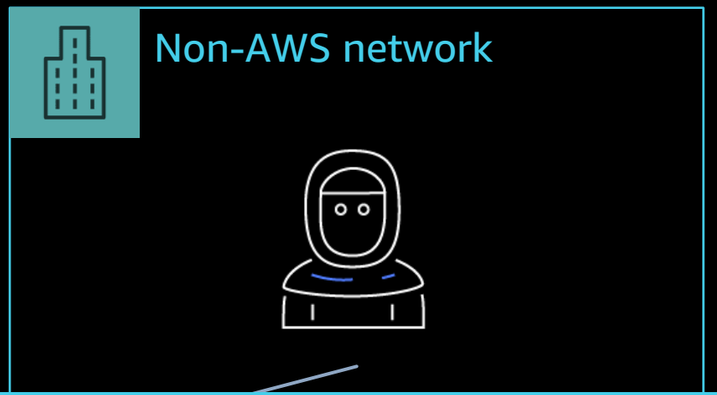
CONFIGURE AND MONITOR NETWORK SECURITY AT SCALE



#7 Assert network origin in your IAM policies

WHEN YOU KNOW THE NETWORK LOCATION FROM WHICH AN IDENTITY WILL BE USED, REQUIRE IT





VPC Endpoint



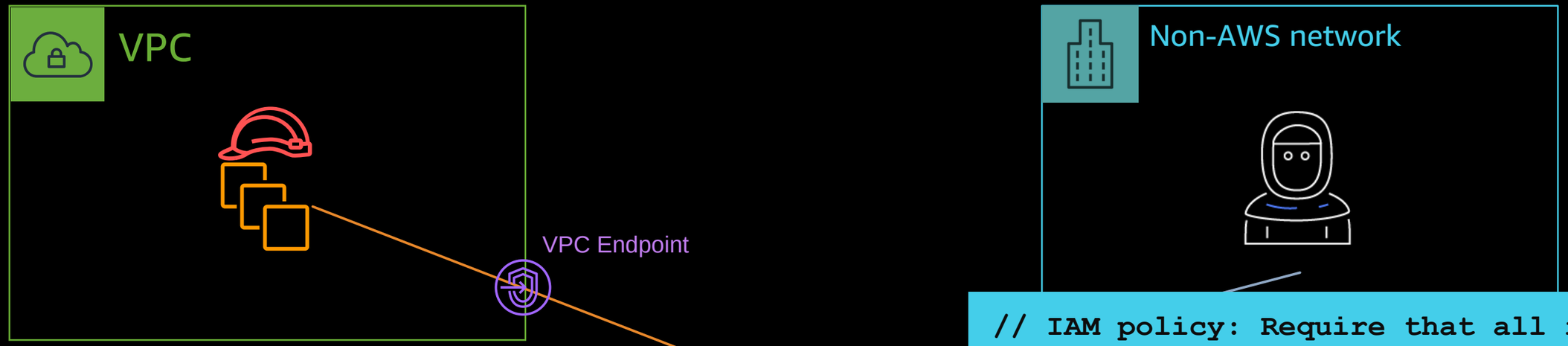
Amazon S3

```
// IAM policy: Require that all requests
// originate from this VPC
{
  "Effect": "Deny",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "StringNotEqualsIfExists": {
      "aws:SourceVpc": "vpc-1234abcd"
```

```
// IAM policy: Require that all requests
originate from this IP address range
{
  "Effect": "Deny",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "NotIpAddress": {
      "aws:SourceIp": "1.2.3.0/24"
    },
    "Bool": {
      "aws:ViaAWSService": "false"
```

#7 Assert network origin in your IAM policies

WHEN YOU KNOW THE NETWORK LOCATION FROM WHICH AN IDENTITY WILL BE USED, REQUIRE IT




```
// IAM policy
// originate
{
  "Effect": "
  "Action": "
  "Resource": "*",
  "Condition": {
    "StringNotEqualsIfExists": {
      "aws:SourceVpc": "vpc-1234abcd"
```

Effective Security Tip #7.5

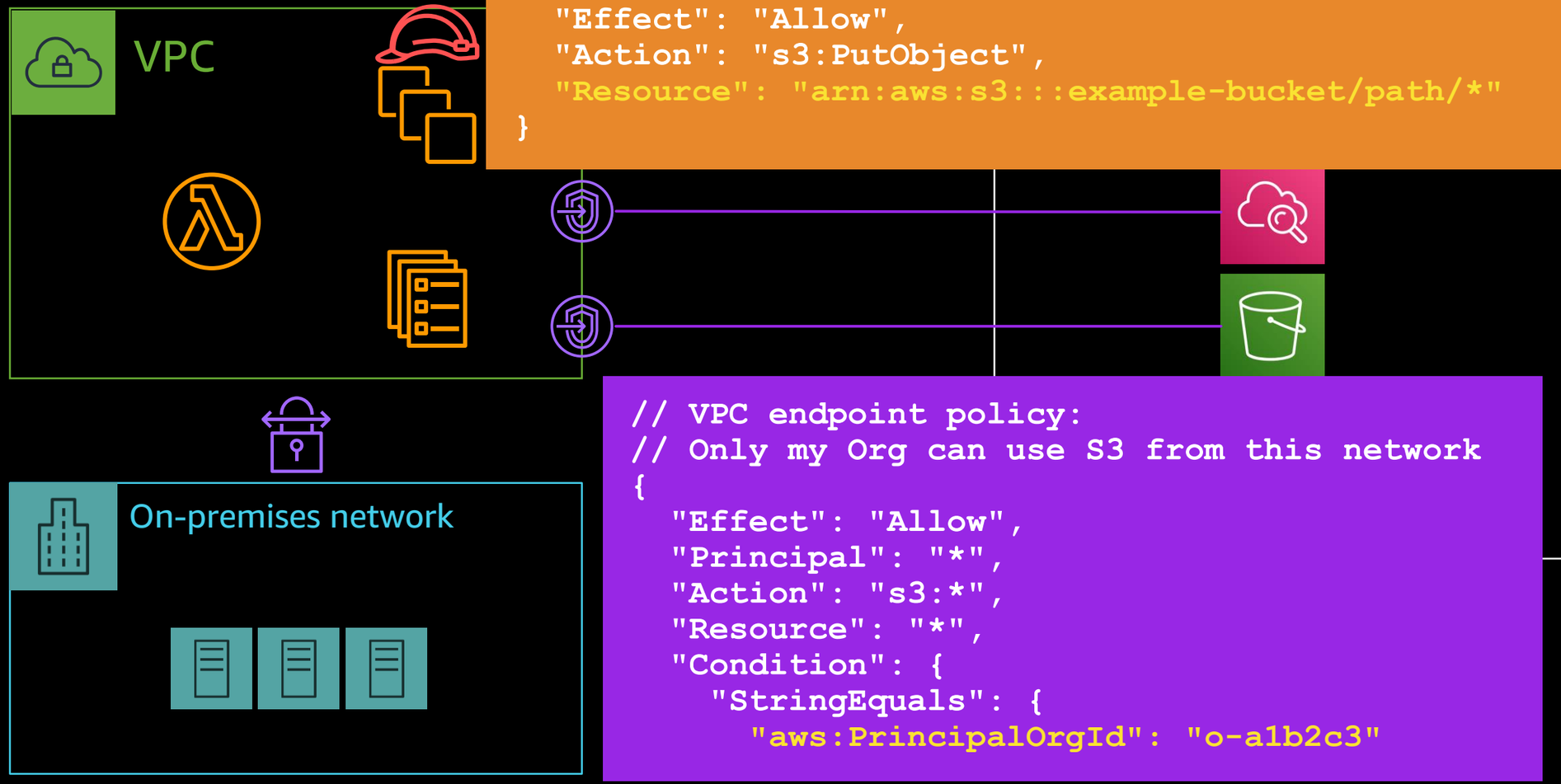
aws:ViaAWSService accounts for intermediate AWS services making API calls on your behalf

```
// IAM policy: Require that all requests
originate from this IP address range

Effect": "Deny",
Action": "*",
Resource": "*",
Condition": {
  "NotIpAddress": {
    "aws:SourceIp": "1.2.3.0/24"
  },
  "Bool": {
    "aws:ViaAWSService": "false"
```

#8 Use the network and IAM to keep your data where you want it

VPC ENDPOINTS LIMIT USE OF YOUR NETWORK TO EXPECTED SCENARIOS; LIMIT THE REST WITH IAM




```
// IAM policy: Allows writes to a specific bucket only
{
  "Effect": "Allow",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::example-bucket/path/*"
}
```

```
// VPC endpoint policy:
// Only my Org can use S3 from this network
{
  "Effect": "Allow",
  "Principal": "*",
  "Action": "s3:*",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:PrincipalOrgId": "o-a1b2c3"
```

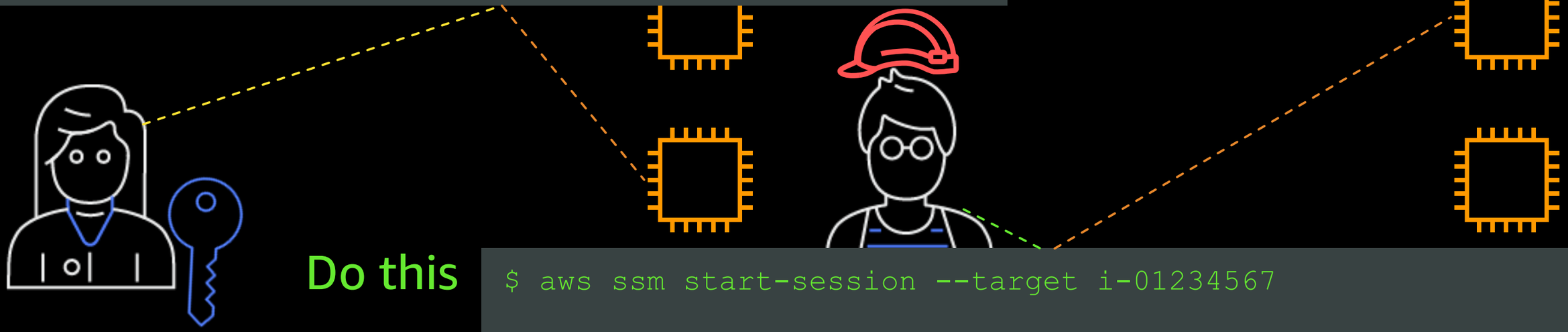
#9 Connect to EC2 instances without SSH keys

AWS SYSTEMS MANAGER SESSION MANAGER AUTHENTICATES AND AUTHORIZE EACH AND EVERY ACCESS

Instead of this

```
$ ssh -i /keys_i_got/something.pem somebody@1.2.3.4

somebody:~$ ./run_big_script.sh ...
```



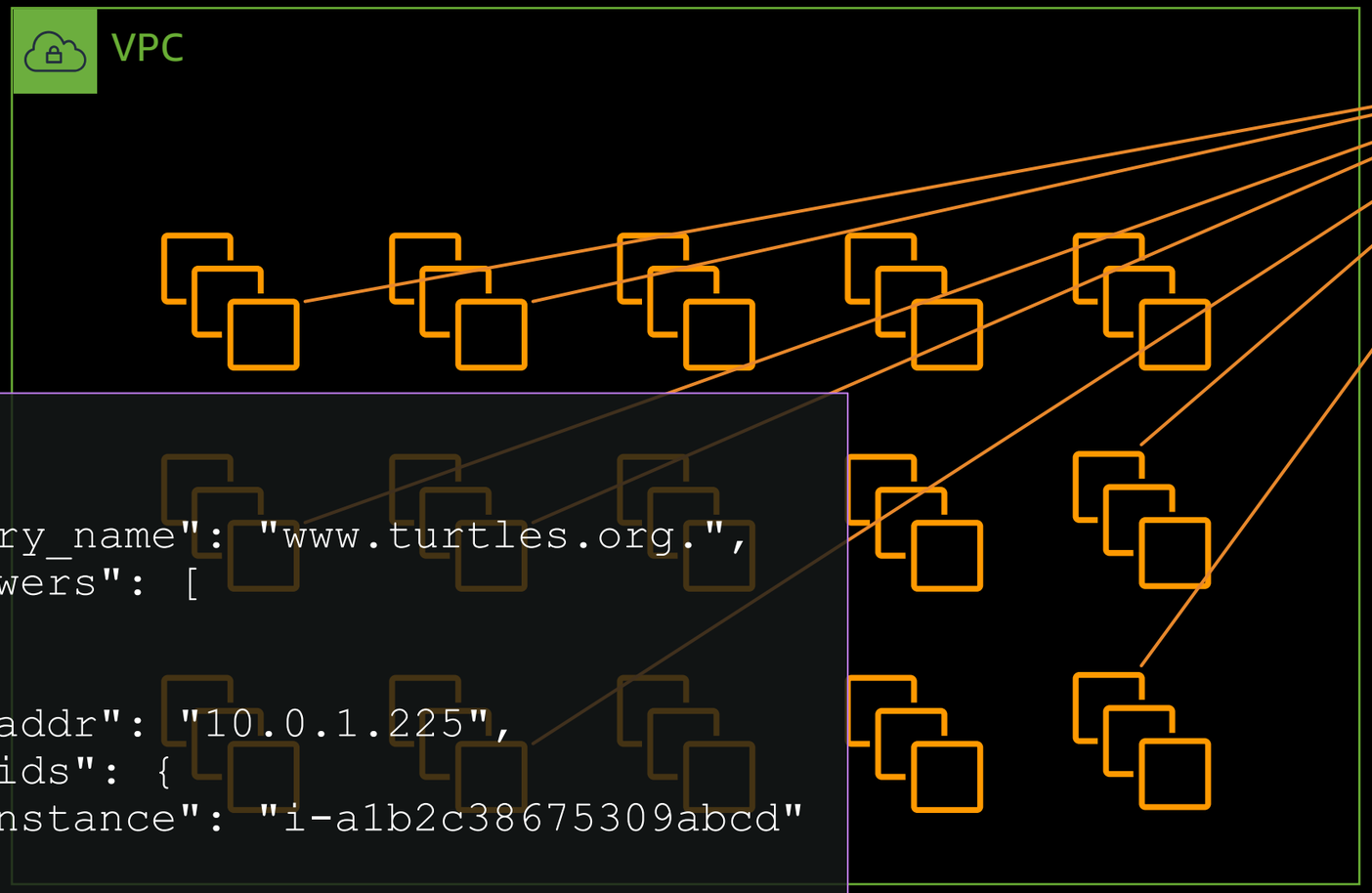
Do this

```
$ aws ssm start-session --target i-01234567

Starting session with SessionId: fred@example.com-04192a...
sh-4.2$
```

#10 Know your DNS: Route 53 Resolver Query Logs

INSIGHTS INTO NETWORK TRAFFIC PATTERNS, FOR SECURITY AND TROUBLESHOOTING





Amazon Route 53



Amazon S3



CloudWatch logs



Amazon Kinesis Data Firehose

```
{
  …
  "query_name": "www.turtles.org.",
  "answers": [
    …
  ],
  "srcaddr": "10.0.1.225",
  "srcids": {
    "instance": "i-a1b2c38675309abcd"
  },
  …
}
```



Here they are again

1. Amazon S3 Block Public Access
2. Use federation – make all your IAM credentials temporary
3. Collect AWS CloudTrail for your entire AWS organization
4. Know how to query a CloudTrail
5. Tag your subnets for scalable control over connectivity
6. Centrally manage network security with AWS Firewall Manager
7. Assert network origin in your AWS Identity and Access Management (IAM) policies
8. Use the network to keep your data where you want
9. Connect to your Amazon Elastic Compute Cloud (Amazon EC2) instances without SSH keys
10. Monitor DNS from your VPC with Amazon Route 53 Resolver Query Logging

Thank you!

Becky Weiss

becky@amazon.com

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.