

# CS 455 – Computer Security Fundamentals

Dr. Chen-Yeou (Charles) Yu

# Computer Security Fundamentals

- Amazon EC2 instances with Security Groups and Session Manager
- AWS Network Access Control List (TBD)
- AWS Web Application Firewall (WAF) (TBD)
  - **AWS WAF Bot Control Explained and Demonstrated**

# Amazon EC2 instances with Security Groups and Session Manager

- First thing of all, let's examine the vulnerabilities of our EC2 instance
- Assuming that, my Kali Linux @ VirtualBox is the hacker's machine
- The instance's public IP address which I get from the AWS EC2 console is:

Public IPv4 address

 18.218.230.188 | [open address](#) 

- Let's do some nmap scans (check the next page)

# Amazon EC2 instances with Security Groups and Session Manager

- I only have port 22 for SSH is open, and

- Port 3389 for RDP is open

- This is the same as our expectations. Remember the inbound traffic?

```
(kali㉿kali)-[~]  
$ nmap 18.218.230.188 -sV -v -Pn  
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-16 22:02 EDT  
NSE: Loaded 45 scripts for scanning.  
Initiating Parallel DNS resolution of 1 host. at 22:02  
Completed Parallel DNS resolution of 1 host. at 22:02, 0.13s elapsed  
Initiating Connect Scan at 22:02  
Scanning ec2-18-218-230-188.us-east-2.compute.amazonaws.com (18.218.230.188) [1000 ports]  
Discovered open port 22/tcp on 18.218.230.188  
Discovered open port 3389/tcp on 18.218.230.188  
Completed Connect Scan at 22:02, 10.22s elapsed (1000 total ports)  
Initiating Service scan at 22:02  
Scanning 2 services on ec2-18-218-230-188.us-east-2.compute.amazonaws.com (18.218.230.188)  
Completed Service scan at 22:02, 11.24s elapsed (2 services on 1 host)  
NSE: Script scanning 18.218.230.188.  
Initiating NSE at 22:02  
Completed NSE at 22:02, 0.01s elapsed  
Initiating NSE at 22:02  
Completed NSE at 22:02, 0.00s elapsed  
Nmap scan report for ec2-18-218-230-188.us-east-2.compute.amazonaws.com (18.218.230.188)  
Host is up (0.067s latency).  
Not shown: 998 filtered tcp ports (no-response)  
PORT      STATE SERVICE      VERSION  
22/tcp    open  ssh          OpenSSH 9.2p1 Debian 2 (protocol 2.0)  
3389/tcp  open  ms-wbt-server xrdp  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
Read data files from: /usr/bin/../../share/nmap  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 21.99 seconds
```

# Amazon EC2 instances with Security Groups and Session Manager

- Let's go back to our EC2 instance.
- Go ahead check the instance, and click the **[Security]** tab
- It seems we do not have IAM setup yet

The screenshot displays the Amazon Management Console interface for an EC2 instance. At the top, there's a header for 'Instances (1/1)' with buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below this is a search bar and a table listing the instance. The instance 'KaliLinux' with ID 'i-0c42e00490f1fa031' is shown in a 'Running' state, using a 't2.micro' instance type, with '2/2 checks passed' and 'No alarms'. The 'Security' tab is selected, showing details for the instance. The 'IAM Role' is set to '-', and the 'Security groups' are 'sg-01d023506013fce97 (Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1)'. The 'Inbound rules' section shows two rules: one for port 3389 (TCP) and one for port 22 (TCP), both with source '0.0.0.0/0'. An orange arrow points to the 'Security' tab, and a blue arrow points to the 'IAM Role' field.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
KaliLinux	i-0c42e00490f1fa031	Running	t2.micro	2/2 checks passed	No alarms	us-east-2c	ec2-18-2

Instance: i-0c42e00490f1fa031 (KaliLinux)

Details | **Security** | Networking | Storage | Status checks | Monitoring | Tags

▼ Security details

IAM Role: -

Owner ID: 378253604690

Launch time: Thu Apr 13 2023 22:03:28 GMT-0500 (Central Daylight Time)

Security groups: sg-01d023506013fce97 (Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1)

▼ Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security group
-	sgr-07ca388d372bb8fc6	3389	TCP	0.0.0.0/0	Kali Linux-Kali
-	sgr-06f0c476645d79813	22	TCP	0.0.0.0/0	Kali Linux-Kali

# Amazon EC2 instances with Security Groups and Session Manager

- Scroll down little bit more, you can see the details for “Security Group”
- Inbound and Outbound rules

The screenshot displays the Amazon Management Console interface for EC2 instances. At the top, there's a header with 'Instances (1/1)' and a search bar. Below this, a table lists the instance details. The instance 'KaliLinux' with ID 'i-0c42e00490f1fa031' is shown in a 'Running' state, using a 't2.micro' instance type. It has passed 2/2 status checks and has no alarms. The instance is located in the 'us-east-2c' availability zone. Below the table, the details for the selected instance are shown, including the security groups it belongs to. The 'Inbound rules' section shows two rules: one for port 3389 (SSH) and another for port 22 (SSH), both allowing traffic from '0.0.0.0/0'. The 'Outbound rules' section shows one rule allowing all traffic to '0.0.0.0/0'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
KaliLinux	i-0c42e00490f1fa031	Running	t2.micro	2/2 checks passed	No alarms	us-east-2c	ec2-18-2...

**Instance: i-0c42e00490f1fa031 (KaliLinux)**

Security groups

- sg-01d023506013fce97 (Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1)

**Inbound rules**

Name	Security group rule ID	Port range	Protocol	Source	Security group
-	sgr-07ca388d372bb8fc6	3389	TCP	0.0.0.0/0	Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1
-	sgr-06f0c476645d79813	22	TCP	0.0.0.0/0	Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1

**Outbound rules**

Name	Security group rule ID	Port range	Protocol	Destination	Security group
-	sgr-0688c44825b4038d8	All	All	0.0.0.0/0	Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1

# Amazon EC2 instances with Security Groups and Session Manager

- Going to the [Security Groups] is easy.
- We can directly go to the security group by using the shortcut, which means the one you are using. Or, go to the link on your LHS

The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, the instance ID `i-0c42e00490f1fa031` is shown with the label `(KaliLinux)`. Below this, the 'Security groups' section lists `sg-01d023506013fce97` with the label `(Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1)`. An orange arrow points from the text 'the one you are using' in the list above to this security group. The 'Inbound rules' section contains a table with two rules:

Name	Security group rule ID	Port range	Protocol	Source	Security group
—	<code>sgr-07ca388d372bb8fc6</code>	3389	TCP	0.0.0.0/0	<a href="#">Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1</a>
—	<code>sgr-06f0c476645d79813</code>	22	TCP	0.0.0.0/0	<a href="#">Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1</a>

The 'Outbound rules' section contains a table with one rule:

Name	Security group rule ID	Port range	Protocol	Destination	Security group
—	<code>sgr-0688c44825b4038d8</code>	All	All	0.0.0.0/0	<a href="#">Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1</a>

On the right side of the console, the 'Network & Security' sidebar is visible, with 'Security Groups' highlighted by an orange arrow pointing from the text 'Or, go to the link on your LHS' in the list above.

# Amazon EC2 instances with Security Groups and Session Manager

After a click on the [Security Groups], we can “tighten” the access rights  
In this case, you can **delete** the whole SSH. That means, no more connection is allowed from SSH. Or change the current setting

The screenshot shows the Amazon EC2 console interface for managing security groups. The top navigation bar includes tabs for 'Inbound rules', 'Outbound rules', and 'Tags'. A notification banner at the top states: 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button. Below this, the 'Inbound rules (2)' section is active, displaying a table of rules. An orange arrow points from the text 'delete the whole SSH' to the checkbox of the SSH rule in this table. Another orange arrow points from the text 'Or change the current setting' to the 'Edit inbound rules' button. A blue arrow points from the 'Edit inbound rules' button to the detailed rule editor below. The detailed editor shows two rules: one for RDP (sgr-07ca388d372bb8fc6) and one for SSH (sgr-06f0c476645d79813). The SSH rule's source is currently set to '0.0.0.0/0'. An orange arrow points from the text 'delete the whole SSH' to the 'Delete' button for the SSH rule. At the bottom, there are buttons for 'Add rule', 'Cancel', 'Preview changes', and 'Save rules'.

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-07ca388d372bb8fc6	IPv4	RDP	TCP	3389	0.0.0.0/0
-	sgr-06f0c476645d79813	IPv4	SSH	TCP	22	0.0.0.0/0

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	Actions
sgr-07ca388d372bb8fc6	RDP	TCP	3389	Custom 0.0.0.0/0		Delete
sgr-06f0c476645d79813	SSH	TCP	22	Custom 0.0.0.0/0		Delete



# Amazon EC2 instances with Security Groups and Session Manager

- What if we only allow local (private) IP addresses to be able to access this EC2 instance? (Allowing connections **within** AWS!)
  - Step1. Delete the current “setting” for SSH (0.0.0.0/0) We took SSH for example
  - Step2. Go back to the EC2 instance and check its Private IPv4 addresses in the [Networking] tab. Then, we get the following: 172.31.41.240

Private IPv4 addresses  
172.31.41.240

Inbound rules [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-07ca388d372bb8fc6	RDP	TCP	3389	Custom	<input type="text"/>	<input type="button" value="Delete"/>
sgr-06f0c476645d79813	SSH	TCP	22	Custom	<input type="text"/>	<input type="button" value="Delete"/>

# Amazon EC2 instances with Security Groups and Session Manager

- Step3. Click the Subnet ID

Subnet ID  
[subnet-065d4b6488ad86539](#)

- It will guide you to this setting,
- Here we go! This is what I want!
- Copy this!

The screenshot shows the AWS Management Console interface for a subnet. The top section displays a table of subnets with columns for Name, Subnet ID, State, VPC, IPv4 CIDR, and IPv6 CIDR. The subnet `subnet-065d4b6488ad86539` is selected and its details are shown below. The details are organized into a grid with four columns: Subnet ID, Subnet ARN, State, and IPv4 CIDR. The IPv4 CIDR field shows `172.31.32.0/20`, which is highlighted by an orange arrow originating from the text 'Copy this!' in the adjacent list.

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
-	subnet-065d4b6488ad86539	Available	vpc-0c9f6edbc269509a5	172.31.32.0/20	-

subnet-065d4b6488ad86539						
Details	Flow logs	Route table	Network ACL	CIDR reservations	Sharing	Tags
<b>Subnet ID</b> subnet-065d4b6488ad86539	<b>Subnet ARN</b> arn:aws:ec2:us-east-2:378253604690:subnet/subnet-065d4b6488ad86539	<b>State</b> Available	<b>IPv4 CIDR</b> 172.31.32.0/20			
<b>Available IPv4 addresses</b> 4090	<b>IPv6 CIDR</b> -	<b>Availability Zone</b> us-east-2c	<b>Availability Zone ID</b> use2-az3			
<b>VPC</b> vpc-0c9f6edbc269509a5	<b>Route table</b> rtb-0c1acffce62c0d6f3	<b>Network ACL</b> acl-064142988a8b509cd	<b>Default subnet</b> Yes			
<b>Auto-assign public IPv4 address</b> Yes	<b>Auto-assign IPv6 address</b> No	<b>Auto-assign customer-owned IPv4 address</b> No	<b>Customer-owned IPv4 pool</b> -			
<b>Outpost ID</b> -		<b>IPv6 CIDR reservations</b> -	<b>IPv6-only</b> No			

# Amazon EC2 instances with Security Groups and Session Manager

- Step4. We can head back to the EC2, inbound rules.
- Paste it to this field
- So, in this way, the IP addresses coming in from this subnet (in the private / local area network) are allowed to use SSH into this EC2 instance. In other words, the IP addresses must be in the AWS as well.
- Step5. Save rules. Then, enjoy your “basic” and “fundamental” protections!

**Inbound rules** [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-07ca388d372bb8fc6	RDP	TCP	3389	Custom	<input type="text"/>	<input type="button" value="Delete"/>
sgr-06f0c476645d79813	SSH	TCP	22	Custom	<input type="text"/>	<input type="button" value="Delete"/>

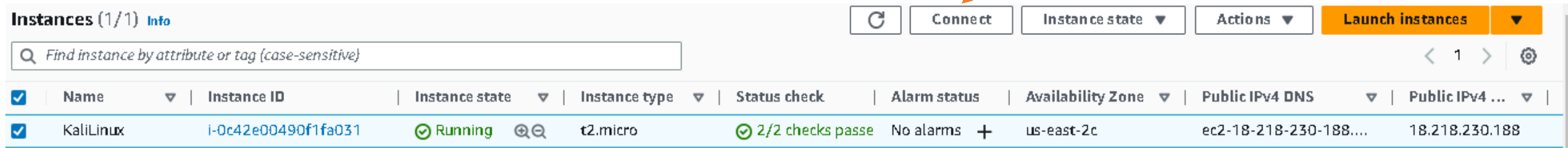
# Amazon EC2 instances with Security Groups and Session Manager

- If I go back to the Kali Linux (VirtualBox) and run the exact same scan?
- See? The SSH port is **closed** to the public internet! (But is open to AWS internal devices!)

```
(kali㉿kali)-[~]  
$ nmap 18.218.230.188 -sV -v -Pn  
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-17 01:14 EDT  
NSE: Loaded 45 scripts for scanning.  
Initiating Parallel DNS resolution of 1 host. at 01:14  
Completed Parallel DNS resolution of 1 host. at 01:14, 0.12s elapsed  
Initiating Connect Scan at 01:14  
Scanning ec2-18-218-230-188.us-east-2.compute.amazonaws.com (18.218.230.188) [1000 ports]  
Discovered open port 3389/tcp on 18.218.230.188  
Completed Connect Scan at 01:14, 10.69s elapsed (1000 total ports)  
Initiating Service scan at 01:14  
Scanning 1 service on ec2-18-218-230-188.us-east-2.compute.amazonaws.com (18.218.230.188)  
Completed Service scan at 01:14, 11.22s elapsed (1 service on 1 host)  
NSE: Script scanning 18.218.230.188.  
Initiating NSE at 01:14  
Completed NSE at 01:14, 0.01s elapsed  
Initiating NSE at 01:14  
Completed NSE at 01:14, 0.00s elapsed  
Nmap scan report for ec2-18-218-230-188.us-east-2.compute.amazonaws.com (18.218.230.188)  
Host is up (0.080s latency).  
Not shown: 999 filtered tcp ports (no-response)  
PORT      STATE SERVICE      VERSION  
3389/tcp  open  ms-wbt-server xrdp  
  
Read data files from: /usr/bin/../../share/nmap  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 22.37 seconds
```

# Amazon EC2 instances with Security Groups and Session Manager

- In the beginning, go to the EC2 instance. Check the instance → Connect.
- You may find the “session manager” is not working



Instances (1/1) [Info](#)

[Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input checked="" type="checkbox"/>	KaliLinux	i-0c42e00490f1fa031	Running	t2.micro	2/2 checks passed	No alarms	us-east-2c	ec2-18-218-230-188...	18.218.230.188

- Check the next page

# Amazon EC2 instances with Security Groups and Session Manager

- It is not working!
- It seems like we haven't finish the IAM setup (check slide #5)
- Step1: Click the [“AWS Systems Manager Quick Setup”](#)

## Connect to instance [Info](#)

Connect to your instance i-0c42e00490f1fa031 (KaliLinux) using any of these options

EC2 Instance Connect

**Session Manager**

SSH client

EC2 serial console



We weren't able to connect to your instance. Common reasons for this include:

1. SSM Agent isn't installed on the instance. You can install the agent on both [Windows instances](#) and [Linux instances](#).
2. The required [IAM instance profile](#) isn't attached to the instance. You can attach a profile using [AWS Systems Manager Quick Setup](#).
3. Session Manager setup is incomplete. For more information, see [Session Manager Prerequisites](#).

## Session Manager usage:

- Connect to your instance without SSH keys or a bastion host.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

# Amazon EC2 instances with Security Groups and Session Manager

- Check your instance info. I know my instance is in us-east-2 as region

Public IPv4 DNS

 ec2-18-218-230-188.us-east-2.compute.amazonaws.com | [open address](#) 

Private IP DNS name (IPv4 only)

 ip-172-31-41-240.us-east-2.compute.internal

- Finish the following setups, click the [Create].
- Bring you to the next screen

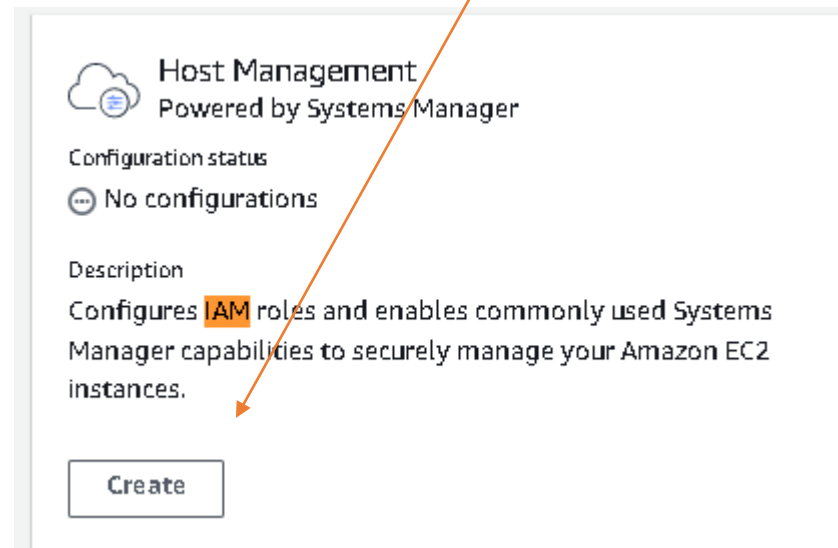
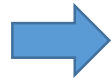
## Get started with Quick Setup

To begin, choose a home AWS Region for Quick Setup. Quick Setup creates the AWS resources used to deploy your configurations in the Region you specify. The home Region can't be changed once chosen.

Choose a home Region

us-east-2 ▼

Get started



# Amazon EC2 instances with Security Groups and Session Manager

- Check the 2 options in Amazon Cloudwatch

## Configuration options

Quick Setup configures the following Systems Manager components based on best practices. Select the check boxes for actions you want to schedule. [Learn more](#)

### Systems Manager

- ☒ Update Systems Manager (SSM) Agent every two weeks.
- ☒ Collect inventory from your instances every 30 minutes.
- ☒ Scan instances for missing patches daily.

### Amazon CloudWatch

- ☒ Install and configure the CloudWatch agent.
- ☒ Update the CloudWatch agent once every 30 days.

If you run this configuration, [Systems Manager Explorer](#) is enabled.

Learn more about the metrics included in [the CloudWatch agent's basic configuration](#) and [Amazon CloudWatch pricing](#).



# Amazon EC2 instances with Security Groups and Session Manager

- Scroll down on the same page and keep everything in the “Targets” to default settings
- Create!

**Targets**  
Targets determine where this configuration is deployed.

Choose between deploying to the current Region or a custom set of Regions.

☒ **Current Region**  
Deploy configuration to the current Region.

☐ **Choose Regions**  
Choose the Regions you want to deploy this configuration to.

Choose how you want to target instances

☒ **All instances**  
Deploy your configuration to all instances in the target account and Regions.

☐ **Tag**  
The key-value pair for the tag you want to target. Specifying a tag selects all instances with that tag.

☐ **Resource group**  
Specify a resource group. Only instances in that group will be configured.

☐ **Manual**  
Manually specify the instances you want to configure.

**Summary**

Choose "Create" to perform the following actions:

- Enable Systems Manager Explorer in all targeted accounts and Regions.
- Deploy IAM roles which enable State Manager to invoke Automation documents that apply selected configuration options.
- Create a State Manager association for each configuration option you have selected.
- Attach instance profiles or IAM roles with required Systems Manager permissions to targeted instances

Cancel Create

# Amazon EC2 instances with Security Groups and Session Manager

- Step2: Install the AWS SSM Agent

## Connect to instance [Info](#)

Connect to your instance i-0c42e00490f1fa031 (KaliLinux) using any of these options

EC2 Instance Connect

**Session Manager**

SSH client

EC2 serial console



We weren't able to connect to your instance. Common reasons for this include:

1. SSM Agent isn't installed on the instance. You can install the agent on both [Windows instances](#) and [Linux instances](#).
2. The required [IAM instance profile](#) isn't attached to the instance. You can attach a profile using [AWS Systems Manager Quick Setup](#).
3. Session Manager setup is incomplete. For more information, see [Session Manager Prerequisites](#).

## Session Manager usage:

- Connect to your instance without SSH keys or a bastion host.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

# Amazon EC2 instances with Security Groups and Session Manager

- By following some tutorial in the internet, make sure the “amazon-ssm-agent” service starts (This is for Debian family)

File Edit Format View Help

Debian:

```
mkdir /tmp/ssm
```

```
wget https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_amd64/amazon-ssm-agent.deb -O /tmp/ssm/amazon-ssm-agent.deb
```

```
sudo dpkg -i /tmp/ssm/amazon-ssm-agent.deb
```

```
sudo service amazon-ssm-agent stop
```

```
sudo -E amazon-ssm-agent -register -code "activation-code" -id "activation-id" -region "region"
```

```
sudo service amazon-ssm-agent start
```

Raspbian:

```
mkdir /tmp/ssm
```

```
sudo curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_arm/amazon-ssm-agent.deb -o /tmp/ssm/amazon-ssm-agent.deb
```

```
sudo dpkg -i /tmp/ssm/amazon-ssm-agent.deb
```

```
sudo service amazon-ssm-agent stop
```

```
sudo -E amazon-ssm-agent -register -code "activation-code" -id "activation-id" -region "region"
```

```
sudo service amazon-ssm-agent start
```

Ubuntu:

```
mkdir /tmp/ssm
```

```
curl https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/debian_amd64/amazon-ssm-agent.deb -o /tmp/ssm/amazon-ssm-agent.deb
```

```
sudo dpkg -i /tmp/ssm/amazon-ssm-agent.deb
```

```
sudo service amazon-ssm-agent stop
```

```
sudo -E amazon-ssm-agent -register -code "activation-code" -id "activation-id" -region "region"
```

```
sudo service amazon-ssm-agent start
```

# Amazon EC2 instances with Security Groups and Session Manager

- This is for CentOS (RedHat) family

```
mkdir /tmp/ssm
cd /tmp/ssm
wget https://s3.ap-south-1.amazonaws.com/amazon-ssm-ap-south-1/latest/linux_amd64/amazon-ssm-agent.rpm
sudo rpm --install amazon-ssm-agent.rpm
sudo systemctl status amazon-ssm-agent
sudo systemctl enable amazon-ssm-agent
sudo systemctl start amazon-ssm-agent
sudo systemctl status amazon-ssm-agent
```

- Of course, our Kali Linux is the Debian family.
- Once you make sure your SSM agent is running, go back to the EC2 instance → [Connect]

# Amazon EC2 instances with Security Groups and Session Manager

- You will see this! Hit the [Connect]

EC2 > Instances > i-0c42e00490f1fa031 > Connect to instance

**Connect to instance** [Info](#)

Connect to your instance i-0c42e00490f1fa031 (KaliLinux) using any of these options

EC2 Instance Connect | **Session Manager** | SSH client | EC2 serial console

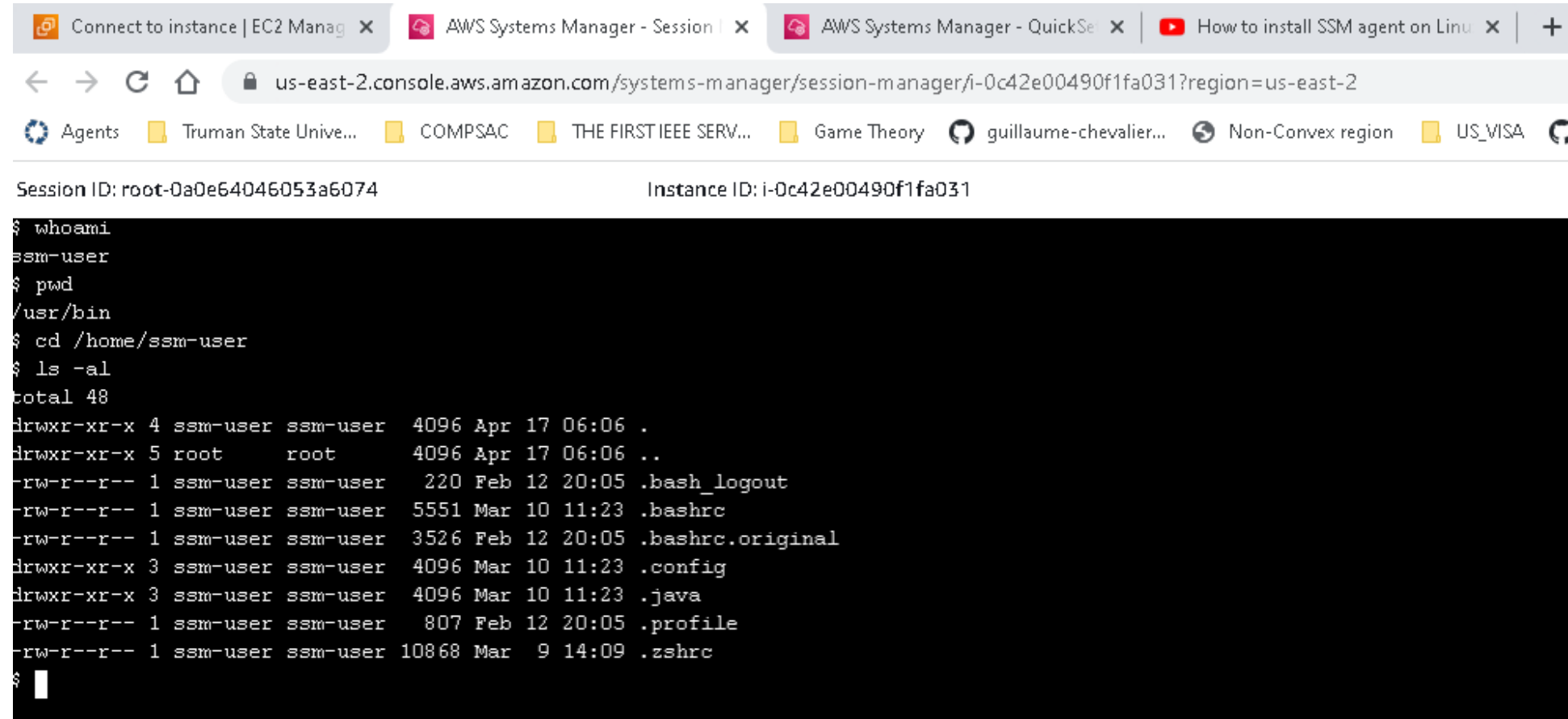
Session Manager usage:

- Connect to your instance without SSH keys or a bastion host.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

Cancel **Connect**

# Amazon EC2 instances with Security Groups and Session Manager

- Here we go! I can access my EC2 instance from the browser!
- SSM Agent!



The screenshot shows a web browser window with the AWS Systems Manager console. The address bar displays the URL: `us-east-2.console.aws.amazon.com/systems-manager/session-manager/i-0c42e00490f1fa031?region=us-east-2`. The browser tabs include "Connect to instance | EC2 Manag...", "AWS Systems Manager - Session...", "AWS Systems Manager - QuickSe...", and "How to install SSM agent on Linu...". The console header shows "Session ID: root-0a0e64046053a6074" and "Instance ID: i-0c42e00490f1fa031". The terminal window shows the following commands and output:

```
$ whoami
ssm-user
$ pwd
/usr/bin
$ cd /home/ssm-user
$ ls -al
total 48
drwxr-xr-x 4 ssm-user ssm-user 4096 Apr 17 06:06 .
drwxr-xr-x 5 root      root    4096 Apr 17 06:06 ..
-rw-r--r-- 1 ssm-user ssm-user  220 Feb 12 20:05 .bash_logout
-rw-r--r-- 1 ssm-user ssm-user 5551 Mar 10 11:23 .bashrc
-rw-r--r-- 1 ssm-user ssm-user 3526 Feb 12 20:05 .bashrc.original
drwxr-xr-x 3 ssm-user ssm-user 4096 Mar 10 11:23 .config
drwxr-xr-x 3 ssm-user ssm-user 4096 Mar 10 11:23 .java
-rw-r--r-- 1 ssm-user ssm-user  807 Feb 12 20:05 .profile
-rw-r--r-- 1 ssm-user ssm-user 10868 Mar  9 14:09 .zshrc
$
```

# Amazon EC2 instances with Security Groups and Session Manager

- Now, we can actually **delete the SSH** form inbound and use the [Session Manager] to manage our system connections.

**Inbound rules** [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-07ca388d372bb8fc6	RDP	TCP	3389	Custom	<input type="text"/>	<input type="button" value="Delete"/>
sgr-06f0c476645d79813	SSH	TCP	22	Custom	<input type="text"/>	<input type="button" value="Delete"/>

- Now we can search the Systems Manager

The screenshot shows the AWS console interface with a search bar at the top containing the text 'systems Manager'. Below the search bar, the search results are displayed. On the left, there is a sidebar with navigation links for 'EC2' and 'Instances'. The main content area is divided into two sections: 'Services' and 'Features'. The 'Services' section lists 'Systems Manager', 'Incident Manager', 'EFS', and 'FSx'. The 'Features' section lists 'Inventory', 'Maintenance Windows', and 'Run Command'. Each item is accompanied by an icon and a brief description.

aws Services

Search results for 'systems Manager'

**Services** [See all 75 results](#)

- Systems Manager** ☆  
AWS Systems Manager is a Central Place to View and Manage AWS Resources
- Incident Manager** ☆  
Automated incident response plans in AWS Systems Manager.
- EFS** ☆  
Managed File Storage for EC2
- FSx** ☆  
Fully managed third-party file systems optimized for a variety of workloads

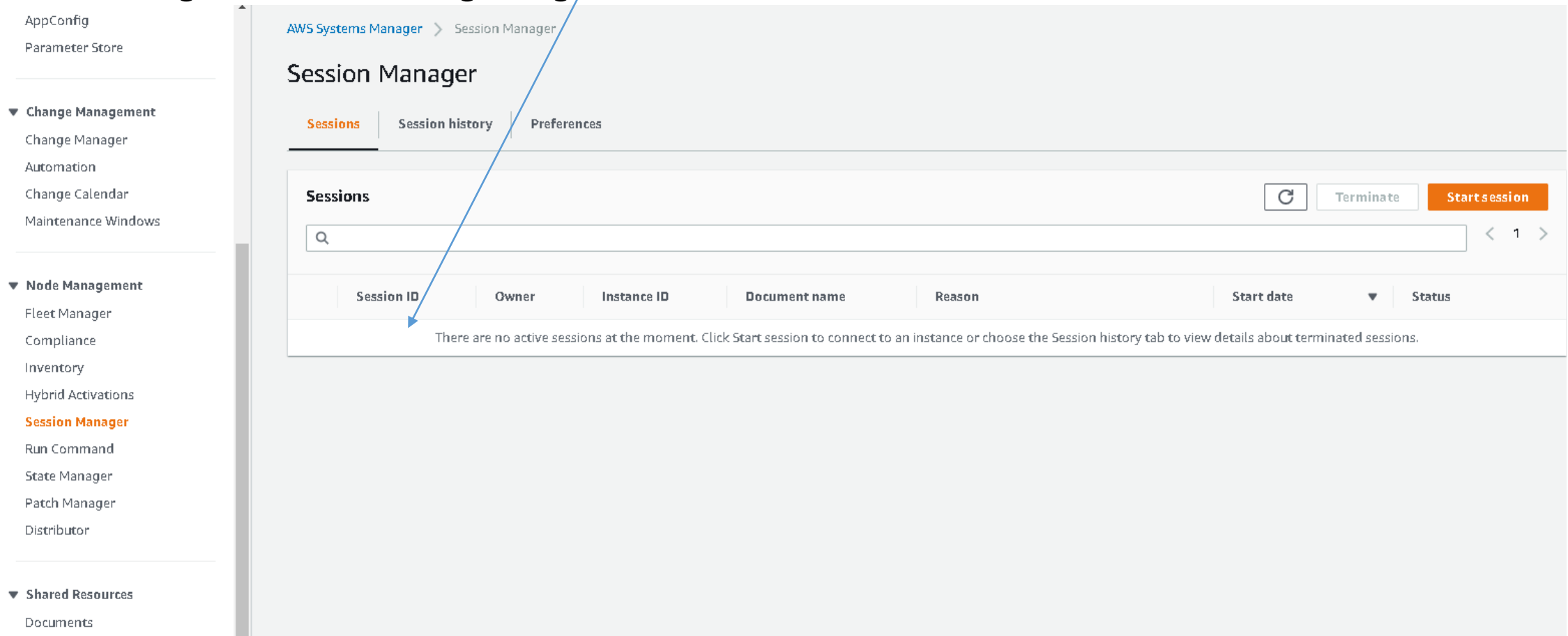
**Features** [See all 95 results](#)

- Inventory**  
 Systems Manager feature
- Maintenance Windows**  
 Systems Manager feature
- Run Command**  
 Systems Manager feature



# Amazon EC2 instances with Security Groups and Session Manager

- You can see right at the bottom left, the session manager
- Nothing is there in the beginning!



The screenshot displays the AWS Systems Manager console interface. On the left is a navigation sidebar with categories: AppConfig, Parameter Store, Change Management (containing Change Manager, Automation, Change Calendar, and Maintenance Windows), Node Management (containing Fleet Manager, Compliance, Inventory, Hybrid Activations, Session Manager, Run Command, State Manager, Patch Manager, and Distributor), and Shared Resources (containing Documents). The 'Session Manager' link under Node Management is highlighted in orange. The main content area is titled 'Session Manager' and has three tabs: 'Sessions' (active), 'Session history', and 'Preferences'. Below the tabs, there's a 'Sessions' section with a search bar and three buttons: a refresh icon, 'Terminate', and 'Start session'. A table with columns 'Session ID', 'Owner', 'Instance ID', 'Document name', 'Reason', 'Start date', and 'Status' is shown, but it is empty. A message at the bottom of the table states: 'There are no active sessions at the moment. Click Start session to connect to an instance or choose the Session history tab to view details about terminated sessions.' A blue arrow points from the text 'Nothing is there in the beginning!' to the empty table.

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

AWS Systems Manager > Session Manager

## Session Manager

**Sessions** | Session history | Preferences

**Sessions** [Refresh] [Terminate] [Start session]

Search

Session ID	Owner	Instance ID	Document name	Reason	Start date	Status
There are no active sessions at the moment. Click Start session to connect to an instance or choose the Session history tab to view details about terminated sessions.						

# Amazon EC2 instances with Security Groups and Session Manager




- If we go back to use the SSM Agent to [Connect] again? So you can do some management jobs over there. You can also check the session and terminate it, if it looks suspicious!

AWS Systems Manager > Session Manager


## Session Manager

**Sessions** | Session history | Preferences

Sessions



< 1 >

	Session ID	Owner	Instance ID	Document name	Reason	Start date ▼	Status
<input type="radio"/>	root-04c4816881ab69b80	arn:aws:iam::378253604690:root	i-0c42e00490f1fa031			Mon, 17 Apr 2023 06:27:23 GMT	 Connected