



# Computing on AWS

## At the core of the lesson

---

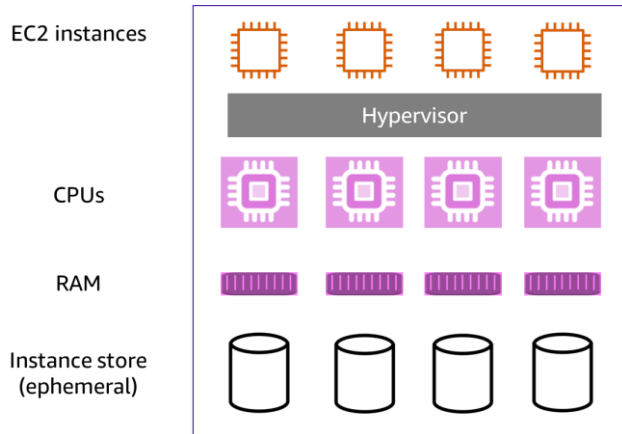
You will learn how to do the following:

- Describe Amazon Elastic Compute Cloud (Amazon EC2) virtualization.
- Identify the steps to launch an EC2 instance.
- Identify best practices for EC2 instances.



# Amazon EC2 virtualization

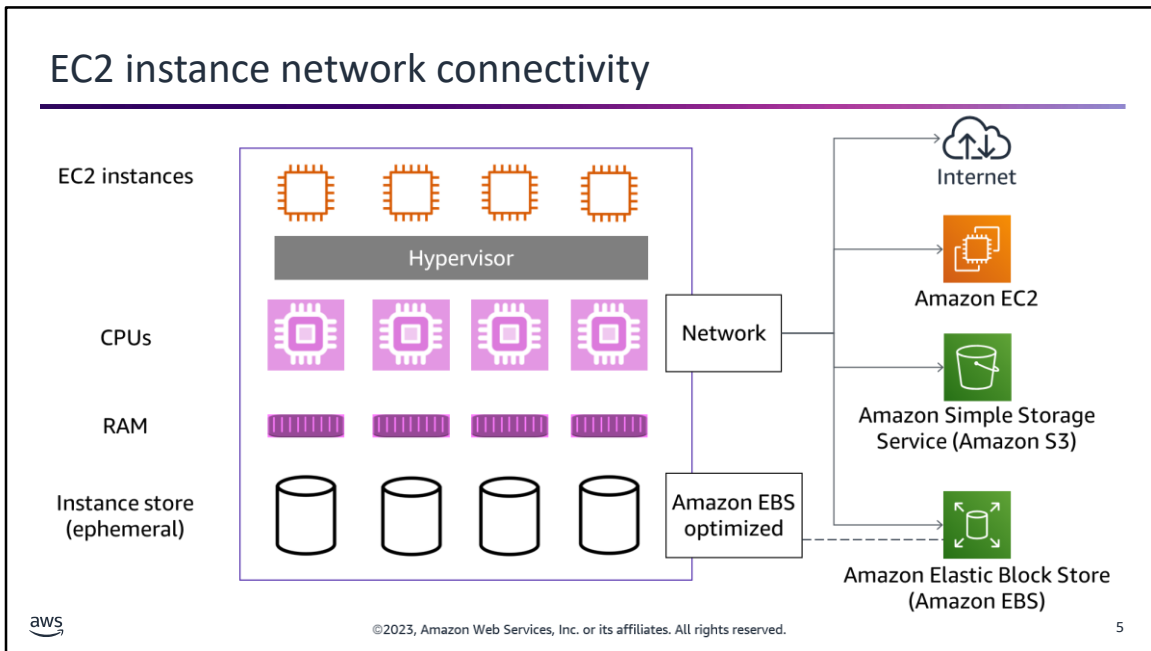
## Amazon EC2 virtualization overview



EC2 instances run as virtual machines on host computers that are located in AWS Availability Zones. Each virtual machine runs an operating system (OS), such as Amazon Linux or Microsoft Windows. You can install and run applications on the OS in each virtual machine or even run enterprise applications that span multiple virtual machines.

The virtual machines run on top of a hypervisor layer that AWS maintains. The hypervisor is the operating platform layer that provides the EC2 instances with access to the actual hardware that the instances need to run. This hardware includes processors, memory, and storage. Each EC2 instance receives a particular number of virtual CPUs for processing and an amount of memory, or RAM.

Some EC2 instances use an instance store. The instance store is also known as ephemeral storage. It is storage that is physically attached to the host computer and provides temporary block-level storage for use with an instance. The data in an instance store persists only during the lifetime of the instance that uses it. If an instance reboots, data in the instance store persists. If the instance stops or terminates, data in the instance store is lost and cannot be recovered.



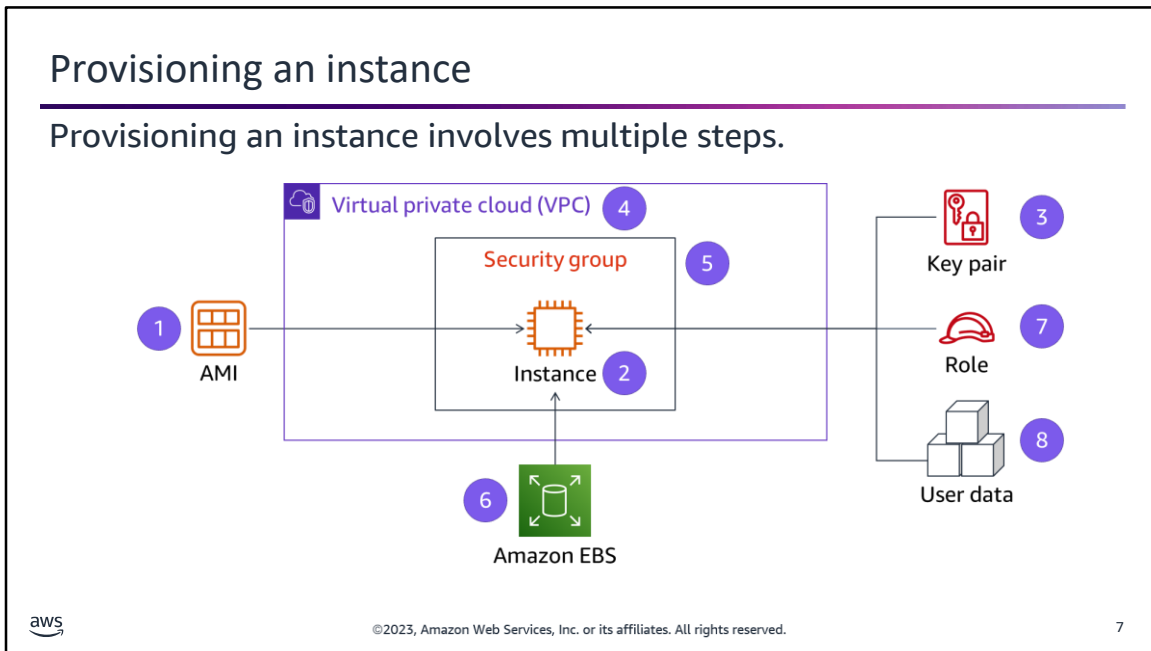
EC2 instances can connect to other resources over a network. For example, many EC2 instances use Amazon Elastic Block Store (Amazon EBS) for the boot disk and other storage needs instead of using an instance store. You attach an EBS volume to an instance through a network connection. Amazon EBS provides persistent block storage volumes, which means that the data will be persisted. For example, the data still persists on the instance even when the instance is in a stopped state.

Amazon EBS optimized instances minimize input/output (I/O) contention between Amazon EBS and other traffic from your instance, which provides better performance. I/O contention occurs when virtual machines compete for I/O resources because there is limited network bandwidth.

EC2 instances can also connect to the internet at large, other EC2 instances, and Amazon Simple Storage Service (Amazon S3) object storage. You can configure the degree of network access to suit your needs and to balance accessibility needs with security requirements. Different instance types provide different levels of network performance.



# Launch an EC2 instance



This diagram illustrates the main steps and components for provisioning an instance. The components that are in the diagram are not an exhaustive list of the options that are available when you configure an instance. However, they are important items that you need for launching a secure instance.

The steps are as follows:

1. You start with an Amazon Machine Image (AMI), which is the template that Amazon EC2 uses to launch an instance. AWS provides some AMIs. Other AMIs come from third-party organizations and are available in the AWS Marketplace. You can also create your own AMI from an existing EC2 instance.
2. After you choose the AMI, you select an instance type. Amazon EC2 provides a selection of instance types that are optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity.
3. If you plan to connect to the instance using Secure Shell (SSH) or Remote Desktop Protocol (RDP), you must specify a key pair. A key pair is a set of security credentials that you use to prove your identity when connecting to an EC2 instance. A key pair consists of a public key and a private key.
4. When you launch an instance, you can specify network placement and addressing as appropriate to secure and provide access to the instance. All instances are deployed within a network either in EC2-Classical or in a virtual private cloud (VPC). You can also decide whether to assign a public IP address or a Domain Name System (DNS) address to the instance.
5. You must also assign a new or existing security group to the instance. A security group is a set of firewall rules that controls the traffic to and from your instance. The security group defines which ports network traffic can use.
6. Next, you specify the storage options for the instance. The storage type that the instance's OS will boot from can be either ephemeral storage or an EBS volume. You can also attach additional block storage volumes to the instance.
7. If you intend to run an application on the instance that makes API calls to AWS services, you must attach an AWS Identity and Access Management (IAM) role to the instance. You use an instance profile to pass an IAM role to an EC2 instance.
8. Finally, you can optionally specify user data when you launch an instance. This data provides a powerful way to automate installations and configurations on the instance when it launches.

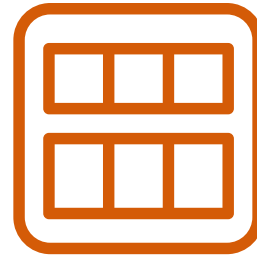
You will learn about these configuration options in detail next.



## AMI

---

- Template that contains information used to create an EC2 instance
- Components:
  - Template for root volume
  - Block device mapping
  - Launch permissions
- Benefits:
  - Repeatable
  - Reusable
  - Available from multiple sources



AMI



An AMI provides the information that is required to launch an instance. You must specify a source AMI when you launch an instance.

An AMI includes a template for the root volume for the instance, which includes an OS, and perhaps an application server and other applications. It also includes a block device mapping that specifies the default EBS volumes and instance store volumes to attach to the instance when it is launched. In addition, the AMI includes launch permissions. These permissions control which AWS accounts can use the AMI. You can also make an AMI available to the public.

You can launch multiple instances from a single AMI when you need multiple instances that have the same configuration. You can also use different AMIs to launch instances when you need instances with different configurations. You can create a new AMI from an instance at any time. For example, you could launch three instances from the same AMI. Then, you can modify each instance in a different way by changing their configurations or installing additional software on some of them. Finally, capture each instance as a new AMI. Thus, you can have three different new AMIs. The changes that you made to each instance are captured as a new template for creating additional matching instances.

You can select an AMI that AWS provides or an AMI that was built by someone in the user community. You can also use an AMI that is available in the AWS Marketplace or one of your own AMIs.

For more information, see “Amazon Machine Images (AMI)” at <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>.

## Instance types

- Instance types are grouped into categories and families and are available for different use cases.
- **t3.nano** is an example of an instance type name.

Category	General purpose	Compute optimized	Memory optimized	Accelerated computing	Storage optimized	High performance computing (HPC) optimized
Instance Family Examples	A1, M4, M5n, Mac, and T3	C4, C5n, C6i, and C7g	R4, R5a, R6in, and X1	F1, G3, Inf1, P2, and VT1	D3, H1, I3, and Im4gn	Hpc6a and Hpc6id
Use Case	Diverse workloads	Compute bound applications	In-memory large data set applications	Hardware accelerated applications	I/O bound applications	HPC workloads



An instance type defines a combination of CPU, memory, storage, and networking capacity. Many instance types exist and give you the flexibility to choose the appropriate mix of resources for your applications. Some are general purpose, and others are designed to provide extra CPU (processing power), extra RAM (memory), or extra I/O network performance. Instance types are grouped by categories and families. You should choose the most cost-effective instance type that supports your workload's requirements. The table in the slide summarizes the different instance type categories, families, and use cases. Note that it provides examples of instance families in each category and not the complete list.

An instance type's name consists of its family name, followed by its generation number, any additional properties, and then its size. For example, the following are the properties for the instance type named t3.nano:

- "t" is the family name.
- "3" is the instance generation.
- "nano" is the instance size.

When you launch a t3.nano instance, it comes with two virtual CPU (vCPU), 0.5 GiB of memory, and up to 5 Gbps of network performance. The OS for a t3.nano instance boots on Amazon EBS storage.

For more information about EC2 instance types, see <https://aws.amazon.com/ec2/instance-types/>.

## Key pairs

- You use a key pair to remotely connect to an instance in a secure manner.
- A key pair consists of the following:
  - A **public key** that AWS stores
  - A **private key** file that you store
- For a Windows AMI, use the private key to obtain the administrator password that you need to log in to your instance.
- For a Linux AMI, use the private key to securely connect to your instance using Secure Shell (SSH).



Data encryption  
key



©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10

Amazon EC2 uses public key cryptography to encrypt and decrypt login information. Public key cryptography uses a public key to encrypt a piece of data, such as a password. The recipient then uses a private key to decrypt the data. The public and private keys are known as a key pair.

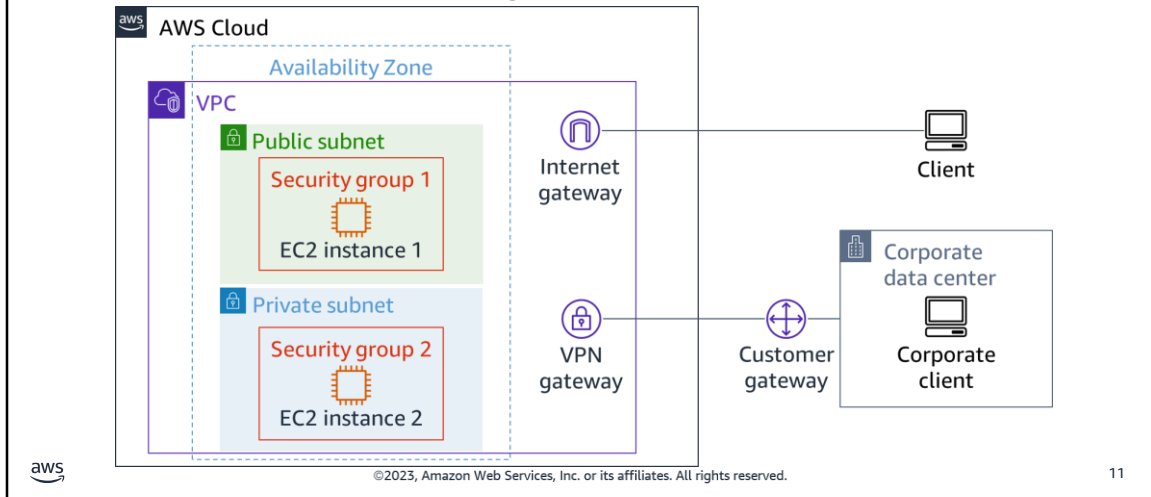
A key pair is necessary to log in to your instance. You need a key pair that is known and registered in the SSH settings of the OS that you are connecting to. Typically, you specify the name of the key pair when you launch the instance. You can create a new key pair and download it as part of the instance launch process. Alternatively, when you launch the instance, you can specify a key pair that you already have access to. When the instance is launched, AWS handles the process of configuring the instance to accept the key pair that you specify. After the instance has booted and you want to connect to it, you can use the private key to connect to the instance.

Default SSH settings on Linux instances do not prompt for a password. Instead, Linux instances expect you to use a key pair to log in although you can configure this process with custom AMIs. With Microsoft Windows instances, you typically use a key pair to decrypt the administrator password and then log in by using Remote Desktop Protocol (RDP).

For more information, see “Amazon EC2 Key Pairs and Linux Instances” at <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>.

# VPC

A VPC provides the networking environment for an EC2 instance.



11

When you launch an EC2 instance, you launch it into a network environment. Typically, you launch it into a VPC that is created with Amazon Virtual Private Cloud (Amazon VPC). The VPC defines a virtual network in your own logically isolated area within the AWS Cloud. You can then launch AWS resources, such as instances, into the VPC. Your VPC closely resembles a traditional network that you might operate in your own data center.

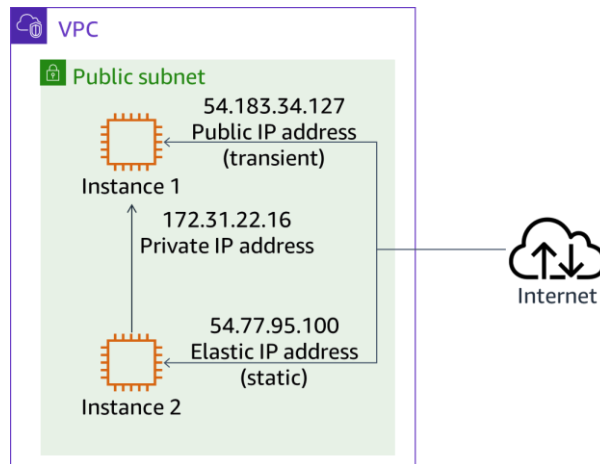
In the VPC, you define one or more subnets. Subnets are logical network segments within the VPC, and each subnet exists within a single Availability Zone. Another part of the network configuration is an internet gateway. An internet gateway is a horizontally scaled, redundant, and highly available VPC component that handles the communication between the instances in your VPC and the internet.

A virtual private gateway is an optional component that supports virtual private network (VPN) connections. The virtual private gateway sits on the Amazon side of the VPN connection. You create a virtual private gateway and attach it to the VPC that you want to create the VPN connection from. The customer side of the VPN connection has a customer gateway, which is a physical device or software application. Notice that the diagram shows only one possible VPN solution.

Security groups are also in this network diagram. Each security group defines a set of firewall rules that allow or block inbound and outbound traffic to or from an instance. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC can be assigned to a different set of security groups. If you do not specify a security group at launch time, the instance will be automatically assigned to the default security group for the VPC.

## Types of IP addresses

- Private IP address
- Public IP address
- Elastic IP address



When you create a new EC2 instance in a VPC, including your default VPC, you must think about the users outside your private network. Should they have access to the instance?

A private IP address is always assigned to each instance when it is launched. It is allocated to the instance from the pool of private IP addresses that are available in the subnet. EC2 instances in the VPC can use private IP addresses to communicate with each other.

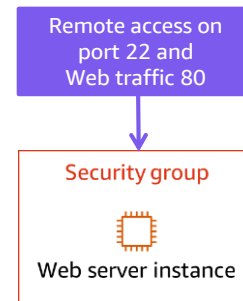
A public IP address can be optionally assigned to an EC2 instance. It is generated dynamically from a pool of available AWS public IP addresses. Clients can use the public IP address to connect to the instance from the internet. If you stop an instance and then start it again, it receives a new public IP address. However, if you reboot an instance, it retains the same public IP address.

An Elastic IP address is a publicly accessible IP address that is allocated from an AWS pool of public IP addresses. An Elastic IP address can optionally be provisioned and then assigned to an EC2 instance. Elastic IP addresses are similar to public IP addresses, except that an Elastic IP address is static. You can reassign an Elastic IP address to another instance at any time.

The diagram shows an example of two EC2 instances in a public subnet of a VPC. An internet client accesses instance 1 using the instance's public IP address of 54.183.34.127. The internet client accesses instance 2 using the instance's Elastic IP address of 54.77.95.100. Instance 2 accesses instance 1 using instance 1's private IP address of 172.31.22.16.

## Security groups

- Restrict access to an instance based on the following:
  - Port range
  - IP address range
  - Resource ID
- Can be associated with multiple instances
- Allow inbound and outbound data
- Can be added or modified after you launch the instance



Each instance must have at least one security group that is associated with it. Security groups are essentially stateful firewalls that surround one or more EC2 instances to give you control over network traffic. A stateful firewall is a firewall that monitors the full state of active network connections. You can control Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) network traffic that can pass to the instance. It's important to understand that security groups are applied to specific instances rather than at the entry point to your network.

In addition to restricting which ports traffic can flow through, you can also restrict which IP addresses that traffic can originate from. If you set the source IP address range as 0.0.0.0/0, traffic on that port will be allowed from any source. However, you can also specify a specific IP address or a Classless Inter-Domain Routing (CIDR) range. Alternatively, you can allow access only from sources within the AWS Cloud that have a specific security group assigned to them. By default, when you create a new security group in a VPC, all outbound traffic is open.

You can assign multiple security groups to a single instance. For example, you can create an administrative security group, which would allow traffic on TCP port 22. You can also create a database server security group, which would allow traffic on TCP port 3306. Then, you can assign both of those security groups to one instance. You can apply a single security group to multiple instances.

## Security group rule examples

Allow HTTP port access from anywhere.	→	<table><tr><th>Rule ID</th><th>Port Range</th><th>Source</th></tr><tr><td>sg-dfc83cba</td><td>80 (HTTP)</td><td>0.0.0.0/0</td></tr></table>	Rule ID	Port Range	Source	sg-dfc83cba	80 (HTTP)	0.0.0.0/0
Rule ID	Port Range	Source						
sg-dfc83cba	80 (HTTP)	0.0.0.0/0						
Allow SSH access from a specific computer.	→	<table><tr><th>Rule ID</th><th>Port Range</th><th>Source</th></tr><tr><td>sg-4ad3712f</td><td>22 (SSH)</td><td>10.50.2.133/32</td></tr></table>	Rule ID	Port Range	Source	sg-4ad3712f	22 (SSH)	10.50.2.133/32
Rule ID	Port Range	Source						
sg-4ad3712f	22 (SSH)	10.50.2.133/32						
Allow SSH access from members of a security group.	→	<table><tr><th>Rule ID</th><th>Port Range</th><th>Source</th></tr><tr><td>sg-d1cd6fb4</td><td>22 (SSH)</td><td>sg-4ad3712f</td></tr></table>	Rule ID	Port Range	Source	sg-d1cd6fb4	22 (SSH)	sg-4ad3712f
Rule ID	Port Range	Source						
sg-d1cd6fb4	22 (SSH)	sg-4ad3712f						



You specify the type of access that a security group allows through a security group rule.

This slide shows three examples of security group rules:

- In the first example, the source description is 0.0.0.0/0. It specifies that any computer from anywhere on the internet can access the instance protected by this security group on port 80. This rule is appropriate, for example, for a web server instance that listens for requests on port 80. Port 80 is the standard HTTP port.
- In the second example, the rule allows traffic from only a specific source CIDR range (10.50.2.133/32) of IP addresses to connect to port 22. Suppose that you want to allow connections to an instance or a set of instances only if they originate from a specific network location. For example, the location might be your office or another subnet in your VPC. In this case, specifying a CIDR range in this way is a good idea.
- In the third example, communication through SSH to instances that belong to the sg-d1cd6fb4 security group is restricted. Only instances that are members of the sg-4ad3712f security group are allowed to do so. This type of rule can be used if you need instances to communicate with one another but want to grant permissions to instances that serve only a particular function in your network. For example, you might have an instance that is the only instance that can be directly connected to from the outside network. This instance, which is often called a bastion host, can be allowed to connect to other instances that are otherwise unreachable from outside the VPC.

## Instance profile

---

- You use an instance profile to attach an AWS Identity and Access Management (IAM) role to an EC2 instance.
- The role supplies temporary permissions that applications running on the instance use to authenticate when they make calls to AWS resources.
- The following are benefits of an instance profile:
  - You don't have to store credentials (access key and secret key) locally on the instance, which is a security risk.
  - Credentials are temporary and rotated automatically.
  - You can use a role for multiple instances (for example, instances in an Auto Scaling group).



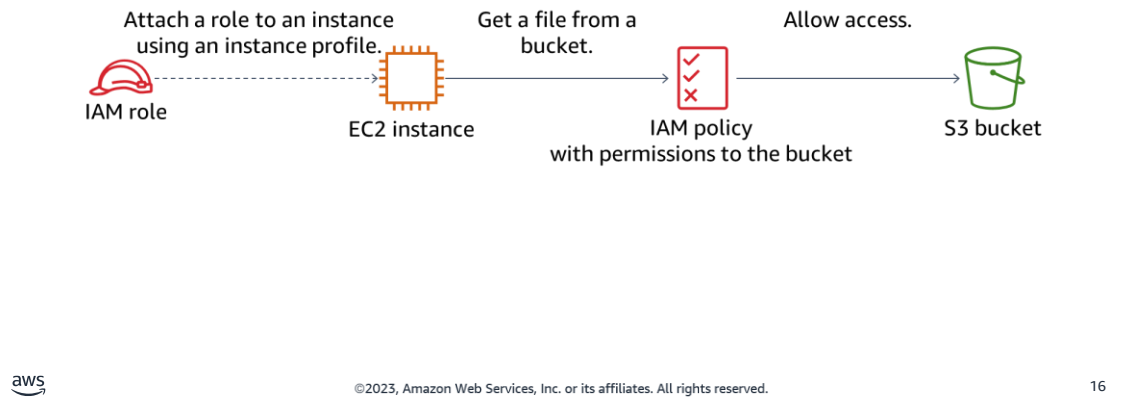
You might want to grant specific access rights for AWS account services to applications that run on an EC2 instance. Or perhaps you want to grant specific access to users who are connected to an EC2 instance. In such cases, you can assign an instance profile to the instance. An instance profile is a container for an IAM role. An IAM role can have one or more IAM policies that are assigned to it, and these policies grant temporary access to account resources.

You can use instance profiles to grant access to specific account resources as needed. Because this functionality exists, you can avoid storing access key and secret access key credentials on EC2 instances. Storing permanent access keys on EC2 instances is not a security best practice, and it should be avoided because credentials can be compromised.



## Instance profile example

You can use an instance profile to grant an application access to an S3 bucket.



This diagram provides an example of how you can use an instance profile to grant permissions to an EC2 instance so that an application running on the instance can access data stored in an S3 bucket. To grant this access, you first define an IAM role with an IAM policy that grants the required level of access to the S3 bucket. Then, add the role to an instance profile, and attach the instance profile to the EC2 instance. You can do this when the instance launches or after it has launched.

At runtime, the application requests to download a file that is stored in the S3 bucket. IAM evaluates the permissions defined in the IAM policy associated with the role. In this case, IAM grants the application access to the file that is stored in Amazon S3.

This example illustrates a useful architectural approach, where essential data files are stored in Amazon S3 instead of being stored locally on an instance. This approach makes the data more globally available and flexible to both access and use.

## User data

---

- You can pass user data to an instance to perform customization and configuration tasks when the instance starts.
- The format of user data varies depending on the OS:
  - A shell script or cloud-init directives on a Linux instance
  - A batch script or a PowerShell script on a Windows instance
- By default, a user data script runs only the first time you launch an instance.



Another useful option that can be invoked when you launch an instance is the user data parameter. With user data, you can supply a script to a Linux or Microsoft Windows instance. This script runs at instance launch as a series of commands during the end of the boot process. User data takes the form of shell scripts or cloud-init directives on Linux instances, or batch or PowerShell scripts on Windows instances. User data scripts are run by the cloud-init service on Linux instances, or by the EC2Launch service on Windows Server instances 2016 or later. By passing user data to an instance, you can automate the setup of a new instance without logging in to the instance.

By default, user data scripts and cloud-init directives run only during the boot cycle when you first launch an instance. However, you can change this configuration so that they run every time the instance is restarted from a stopped state.

## User data on Linux example

### Shell script

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y php7.2
yum install -y httpd
systemctl start httpd
systemctl enable httpd
```

### Cloud-init directives

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
- httpd

runcmd:
- [ sh, -c, "amazon-linux-extras install -y php7.2" ]
- systemctl start httpd
- sudo systemctl enable httpd
```



User data on Linux can take one of two forms: shell script or cloud-init directives. Note that the format of cloud-init directives is YAML.

In this example, the shell script and cloud-init directives versions of the user data perform the same tasks as follows:

1. Update the distribution software packages on the instance.
2. Install the php package.
3. Install the httpd (web server) service.
4. Start the httpd service.
5. Configure the httpd service so that it starts automatically whenever the instance is started.

## EC2 instance metadata

---

- Instance metadata is data about a running EC2 instance.
- Instance metadata is divided in categories, including the following:
  - instance-id
  - instance-type
  - ami-id
  - public-hostname
- To retrieve instance metadata from within the instance, use the following URL: <http://169.254.169.254/latest/meta-data/>
- You can query instance metadata from a user data script to retrieve properties that you can use in the script.



©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

19

The descriptions of the instance metadata categories listed on this slide are as follows:

- instance-id: The ID of the instance
- instance-type: The type of instance
- ami-id: The AMI ID used to launch the instance
- public-hostname: The instance's public DNS (IPv4) hostname if one was assigned to it

To retrieve instance metadata from a command line session within an instance, use the following command:

```
curl http://169.254.169.254/latest/meta-data/
```

## Using instance metadata in a user data script example

```
#!/bin/bash

# Get the hostname from instance metadata.
newHost=$(curl http://169.254.169.254/latest/meta-data/hostname/)

# Change the hostname in /etc/hosts & /etc/sysconfig/network.
sudo sed -i "s/\<localhost\>/$newHost/g" /etc/hosts
sudo sed -i "s/\<localhost\>/$newHost/g" /etc/sysconfig/network

# Reboot the instance.
sudo reboot
```

Query the hostname  
from the instance  
metadata.

Change the hostname in  
the operating system (OS)  
configuration files.



An example of where you could use EC2 instance metadata is in a user data script. In the user data script, you can reference the properties of the EC2 instance, such as hostname, which are not known until launch time. You can retrieve the value of an instance metadata category and assign it to a variable in the user data script.

In the example, the script queries the instance metadata to return the instance's hostname and assigns the value returned to the `newHost` variable. The script then replaces every occurrence of "localhost" to the `newHost` value in the `/etc/hosts` and `/etc/sysconfig/network` OS configuration files. The system is then rebooted to help ensure that the modified OS files values are propagated.


For more information about instance metadata and user data, see <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html>.

## Retrieving instance user data

You can retrieve the user data from a running instance by using the following URL:

<http://169.254.169.254/latest/user-data>

Example of using the curl command to access the user data URL.

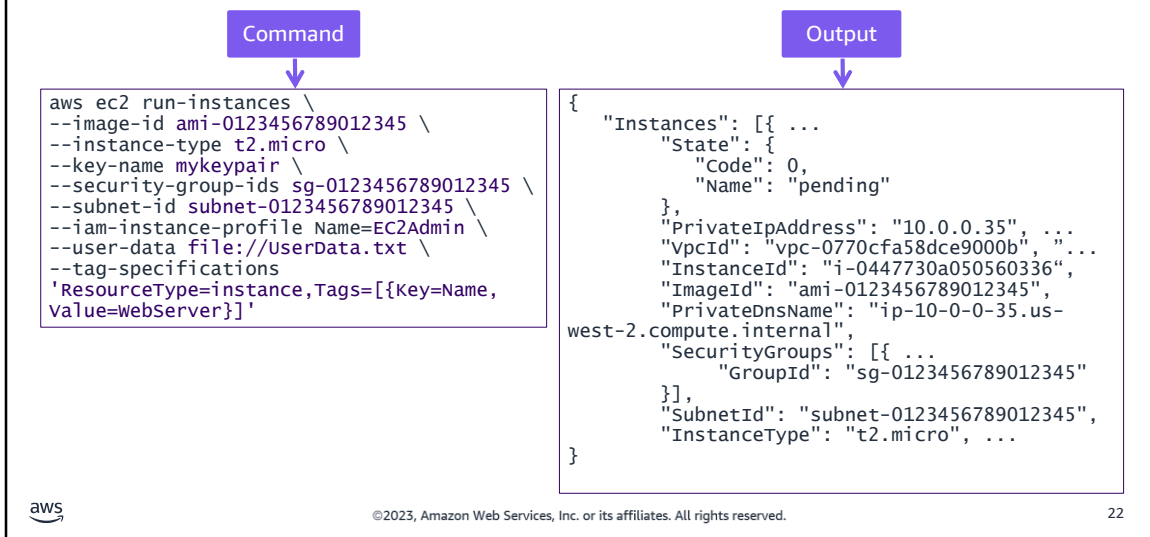


```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

If you want to see the user data that was passed to an instance when it was launched, access the following URL from within instance: <http://169.254.169.254/latest/user-data>

The example on this slide shows how to use the curl command to access the user data URL and retrieve the instance user data.

## Launching an instance using the AWS CLI



This example shows the AWS Command Line Interface (AWS CLI) command that launches an EC2 instance. Notice all the parameters that are specified in the command, most of which were described in detail in the previous slides. The tag-specifications parameter gives you the ability to apply tags, as a name-value pair, to the instance.

You can specify many parameters and options when you run the run-instances command.

For more information, see the AWS CLI Command Reference page for the run-instances command at <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/run-instances.html>.



# Best practices for EC2 instances



## Instance security

---

- Protect the default user account (**ec2-user** on Linux and **Administrator** on Windows) because it has administrative permissions.
- Create additional accounts for new users to access the instance.
  - Create a key pair or use an existing key pair for the new user.
  - For a Linux instance, add new user accounts with SSH access to the instance, and do not use password logins.
  - For a Windows instance, use Active Directory or AWS Directory Service to tightly and centrally control user and group access.
- Apply security patches regularly.



The default `ec2-user` credentials on Linux instances or default administrator username and password for Microsoft Windows instances are starting points to secure your instance. In addition, you must define your own instance-level security and decide how to grant and revoke permissions for instance access as people join and leave your organization. If multiple users require access to the instance, it's a security best practice to use separate accounts for each user.

You can use Amazon EC2 to create a key pair, or you can use a third-party tool to create a key pair and then import the public key to Amazon EC2. For example, you can use a tool such as `ssh-keygen` to generate a public key and private key pair for users who must access the instance.

For a Linux instance, you should use secure and encrypted protocols such as SSH to access your instance because passwords are vulnerable to basic security attacks. For Microsoft Windows instances, use AWS Directory Service to grant and revoke access to machines based on existing Windows users and groups.

Finally, it's important to always stay current with the latest OS updates and security patches. Remember to apply these updates regularly on your instance.

## Remote connection to an instance

Use **EC2 Instance Connect** or **Session Manager**, a capability of AWS Systems Manager, to connect to your instances without the need to manage SSH keys.

Instance Remote Connect Tool	Features
EC2 Instance Connect	<ul style="list-style-type: none"><li>• Supports Amazon Linux 2 and Ubuntu instances</li><li>• Can be accessed through the AWS Management Console</li><li>• Uses IAM policies to control user access to an instance</li><li>• Requires opening an SSH port on the instance</li></ul>
Session Manager	<ul style="list-style-type: none"><li>• Supports Linux, Windows, and macOS instances</li><li>• Can be accessed through the AWS Management Console</li><li>• Uses IAM policies to control user access to an instance</li><li>• Does not require opening an SSH port on the instance</li><li>• Does not require a key pair</li></ul>



The recommended way to remotely connect to an EC2 instance is to use either EC2 Instance Connect or Session Manager, a capability of AWS Systems Manager. One of the main benefits of using these tools is that they are directly accessible through the AWS Management Console using a browser-based shell. They do not require you to install an SSH or RDP client to access an instance. Both tools also do not require you to manage SSH keys.

EC2 Instance Connect provides a secure way to connect to an Amazon Linux 2 or Ubuntu instance using SSH. You can use EC2 Instance Connect to connect to an instance using the Amazon EC2 console, the EC2 Instance Connect CLI, or an SSH client of your choice. With EC2 Instance Connect, you use IAM policies to control the SSH access to the instance. In addition, EC2 Instance Connect logs all connection requests to AWS CloudTrail so that you can audit connection requests.

Session Manager provides the ability to connect to a Linux, Windows, or macOS instance through the AWS Management Console or through the AWS CLI. It provides secure and auditable node management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys. Session Manager also logs connection requests to CloudTrail.

## Additional best practices

---

- Use the instance console screenshot capability to troubleshoot launch or remote connection problems.
- Turn on termination protection to protect an instance from accidental termination.
- Turn off source and destination check on a network address translation (NAT) instance.



For visibility into the state of an instance, AWS provides the ability to generate a screen capture of the instance console from the AWS Management Console. You can generate screenshots while the instance is running or after it has crashed. The generated image is in .jpg format and can be downloaded to a local computer. This capability can help troubleshoot problems, such as when an instance becomes unreachable through a remote connection.

To help protect against data loss that is caused from accidental termination of an EC2 instance, consider turning on termination protection. Termination protection prevents an instance from being accidentally terminated by requiring that termination protection first be turned off before the instance can be terminated.

Each EC2 instance performs source and destination checks by default. Thus, the instance must be the source or destination of any traffic that it sends or receives. However, a network address translation (NAT) instance must be able to send and receive traffic when the source or destination is not itself. Therefore, you must turn off source and destination checks on a NAT instance.

## Checkpoint questions

---

1. What is an AMI?
2. What is the purpose of the user data parameter?
3. An administrator launches a new EC2 instance that starts successfully. The instance shows that it has a public IP address. However, when a user connects with SSH, an error occurs that indicates that the operation timed out. What could the problem be?
4. Which IP address is used to access the metadata by a script that is running on an instance?



The answers to the questions are as follows:

1. What is an AMI?  
An Amazon Machine Image (AMI) provides the information that is needed to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.
2. What is the purpose of the user data parameter?  
User data can be used to perform common automated configuration tasks by running scripts when the instance starts. You can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives.
3. An administrator launches a new EC2 instance that starts successfully. The instance shows that it has a public IP address. However, when a user connects with SSH, an error occurs that indicates that the operation timed out. What could the problem be?  
It is likely that the security group for the instance is not allowing traffic inbound to port 22.
4. Which IP address is used to access the metadata by a script that is running on an instance?  
Metadata is accessed through the following IP address: 169.254.169.254

## Key ideas

---



- EC2 instances are virtual machines that can connect to other resources over a network.
- Launching an instance involves multiple steps, including the following:
  - Selecting an AMI and an instance type
  - Creating a key pair and attaching an instance profile
  - Selecting a VPC and assigning a security group
  - Specifying storage options and user data
- Use EC2 Instance Connect or Session Manager to remotely connect to an instance.



# Thank you

Corrections, feedback, or other questions?  
Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>.  
All trademarks are the property of their owners.