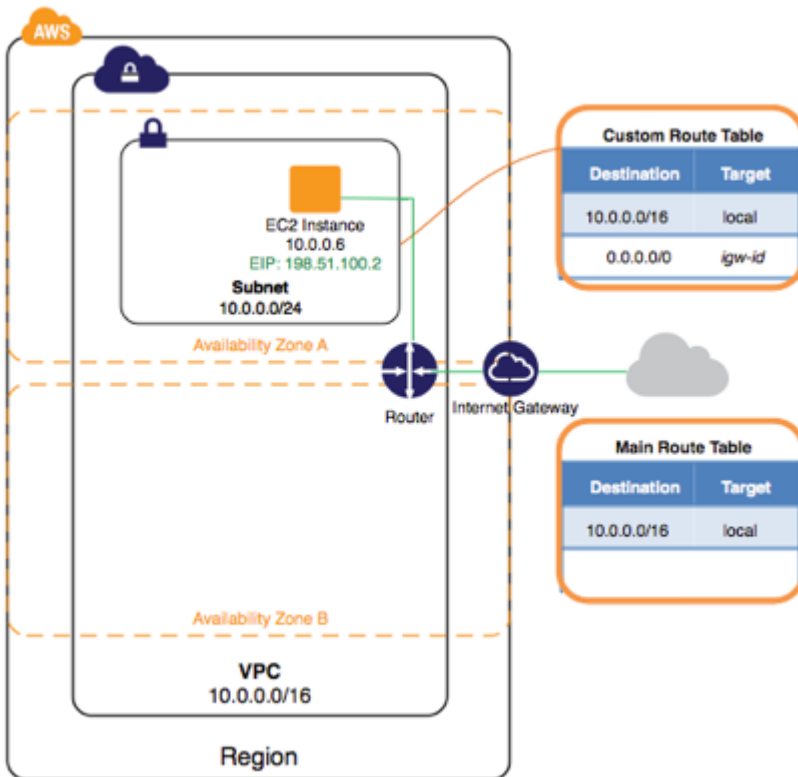
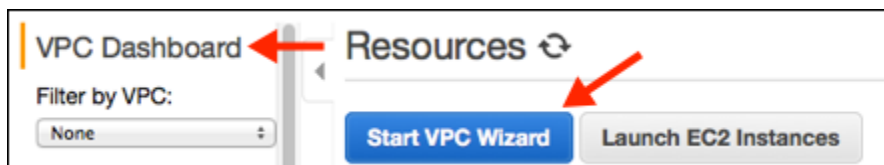


Create a Web Server Instance in VPC



To create a VPC using the Amazon VPC Wizard

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation bar, on the top-right, take note of the region in which you'll be creating the VPC. Ensure that you continue working in the same region for the rest of this exercise, as you cannot launch an instance into your VPC from a different region. For more information about regions, see [Regions and Availability Zones](#).
3. In the navigation pane, choose **VPC dashboard**, and then choose **Start VPC Wizard**.



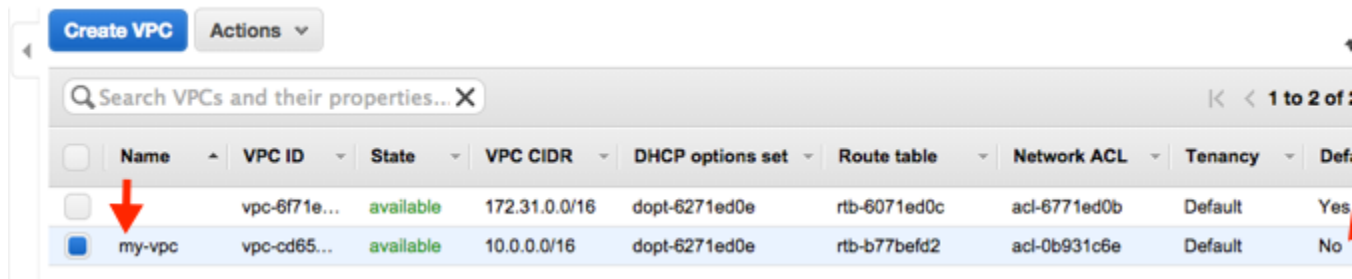
Note

Do not choose **Your VPCs** in the navigation pane; you cannot access the VPC wizard from this page.

4. Choose the first option, **VPC with a Single Public Subnet**, and then choose **Select**.
5. On the configuration page, enter a name for your VPC in the **VPC name** field; for example, `my-vpc`, and enter a name for your subnet in the **Subnet name** field. This helps you to identify the VPC and subnet in the Amazon VPC console after you've created them. For this exercise, you can leave the rest of the configuration settings on the page, and choose **Create VPC**.

(Optional) If you prefer, you can modify the configuration settings as follows, and then choose **Create VPC**.

- The **IP CIDR block** displays the IP address range that you'll use for your VPC (`10.0.0.0/16`), and the **Public subnet** field displays the IP address range you'll use for the subnet (`10.0.0.0/24`). If you don't want to use the default CIDR ranges, you can specify your own.
 - The **Availability Zone** list enables you to select the Availability Zone in which to create the subnet. You can leave **No Preference** to let AWS choose an Availability Zone for you
 - In the **Add endpoints for S3 to your subnets** section, you can select a subnet in which to create a VPC endpoint to Amazon S3 in the same region.
 - The **Enable DNS hostnames** option, when set to **Yes**, ensures that instances that are launched into your VPC receive a DNS hostname.
 - The **Hardware tenancy** option enables you to select whether instances launched into your VPC are run on shared or dedicated hardware. Selecting a dedicated tenancy incurs additional costs.
6. A status window shows the work in progress. When the work completes, choose **OK** to close the status window.
 7. The **Your VPCs** page displays your default VPC and the VPC that you just created. The VPC that you created is a non default VPC, therefore the **Default VPC** column displays **No**.



Buttons: Create VPC, Actions

Search: Search VPCs and their properties... X

Page: 1 to 2 of 2

<input type="checkbox"/>	Name	VPC ID	State	VPC CIDR	DHCP options set	Route table	Network ACL	Tenancy	Def
<input type="checkbox"/>		vpc-6f71e...	available	172.31.0.0/16	dopt-6271ed0e	rtb-6071ed0c	acl-6771ed0b	Default	Yes
<input checked="" type="checkbox"/>	my-vpc	vpc-cd65...	available	10.0.0.0/16	dopt-6271ed0e	rtb-b77befd2	acl-0b931c6e	Default	No

Viewing Information About Your VPC

After you've created the VPC, you can view information about the subnet, the Internet gateway, and the route tables. The VPC that you created has two route tables — a main route table that all VPCs have by default, and a custom route table that was created by the wizard. The custom route table is associated with your subnet, which means that the routes in that table determine how the traffic for the subnet flows. If you add a new subnet to your VPC, it uses the main route table by default.

To view information about your VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Your VPCs**. Take note of the name and the ID of the VPC that you created (look in the **Name** and **VPC ID** columns). You will use this information to identify the components that are associated with your VPC.
3. In the navigation pane, choose **Subnets**. The console displays the subnet that was created when you created your VPC. You can identify the subnet by its name in **Name** column, or you can use the VPC information that you obtained in the previous step and look in the **VPC** column.
4. In the navigation pane, choose **Internet Gateways**. You can find the Internet gateway that's attached to your VPC by looking at the **VPC** column, which displays the ID and the name (if applicable) of the VPC.
5. In the navigation pane, choose **Route Tables**. There are two route tables associated with the VPC. Select the custom route table (the **Main** column displays **No**), and then choose the **Routes** tab to display the route information in the details pane:
 - The first row in the table is the local route, which enables instances within the VPC to communicate. This route is present in every route table by default, and you can't remove it.

- The second row shows the route that the Amazon VPC wizard added to enable traffic destined for an IP address outside the VPC (0.0.0.0/0) to flow from the subnet to the Internet gateway.
6. Select the main route table. The main route table has a local route, but no other routes.

Creating Your WebServerSG Security Group

You can create your security group using the Amazon VPC console.

To create the WebServerSG security group and add rules

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.
4. In the **Group name** field, enter `WebServerSG` as the name of the security group, and provide a description. You can optionally use the **Name tag** field to create a tag for the security group with a key of `Name` and a value that you specify.
5. Select the ID of your VPC from the **VPC** menu, and then choose **Yes, Create**.
6. Select the `WebServerSG` security group that you just created (you can view its name in the **Group Name** column).
7. On the **Inbound Rules** tab, choose **Edit** and add rules for inbound traffic as follows, and then choose **Save** when you're done:
 - a. Select **HTTP** from the **Type** list, and enter `0.0.0.0/0` in the **Source** field.
 - b. Choose **Add another rule**, then select **HTTPS** from the **Type** list, and enter `0.0.0.0/0` in the **Source** field.
 - c. Choose **Add another rule**. If you're launching a Linux instance, select **SSH** from the **Type** list, or if you're launching a Windows instance, select **RDP** from the **Type** list. Enter your network's public IP address range in the **Source** field. If you don't know this address range, you can use `0.0.0.0/0` for this exercise.

Caution

If you use `0.0.0.0/0`, you enable all IP addresses to access your instance using SSH or RDP. This is acceptable for the short exercise, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

Type	Protocol	Port Range	Source	Remove
HTTP (80)	TCP (6)	80	0.0.0.0/0	✗
HTTPS (443)	TCP (6)	443	0.0.0.0/0	✗
SSH (22)	TCP (6)	22	192.0.2.0/24	✗
RDP (3389)	TCP (6)	3389	192.0.2.0/24	✗

Add another rule

To launch an EC2 instance into a VPC

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation bar, on the top-right, ensure that you select the same region in which you created your VPC and security group.
3. From the dashboard, choose **Launch Instance**.
4. On the first page of the wizard, choose the AMI that you want to use. For this exercise, we recommend that you choose an Amazon Linux AMI or a Windows AMI.
5. On the **Choose an Instance Type** page, you can select the hardware configuration and size of the instance to launch. By default, the wizard selects the first available instance type based on the AMI you selected. You can leave the default selection, and then choose **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, select the VPC that you created from the **Network** list, and the subnet from the **Subnet** list. Leave the rest of the default settings, and go through the next pages of the wizard until you get to the **Tag Instance** page.
7. On the **Tag Instance** page, you can tag your instance with a `Name` tag; for example `Name=MyWebServer`. This helps you to identify your instance in the

Amazon EC2 console after you've launched it. Choose **Next: Configure Security Group** when you are done.

8. On the **Configure Security Group** page, the wizard automatically defines the launch-wizard-x security group to allow you to connect to your instance. Instead, choose the **Select an existing security group** option, select the **WebServerSG** group that you created previously, and then choose **Review and Launch**.
9. On the **Review Instance Launch** page, check the details of your instance, and then choose **Launch**.
10. In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. If you create a new key pair, ensure that you download the file and store it in a secure location. You'll need the contents of the private key to connect to your instance after it's launched.

To launch your instance, select the acknowledgment check box, and then choose **Launch Instances**.

11. On the confirmation page, choose **View Instances** to view your instance on the **Instances** page. Select your instance, and view its details in the **Description** tab. The **Private IPs** field displays the private IP address that's assigned to your instance from the range of IP addresses in your subnet.

To allocate and assign an Elastic IP address

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate New Address**, and then **Yes, Allocate**.

Note

If your account supports EC2-Classic, first select **EC2-VPC** from the **Network platform** list.

4. Select the Elastic IP address from the list, choose **Actions**, and then choose **Associate Address**.

5. In the dialog box, choose **Instance** from the **Associate with** list, and then select your instance from the **Instance** list. Choose **Yes, Associate** when you're done.

A Linux Server with httpd installed and deploy a index.html
Connect to the Server console

```
# yum install httpd24
#service httpd start
#chkconfig httpd on
#vi /var/www/html
Esc i
Test webpage
Esc:wq!
```

6. Test the Web Page accessible in Internet