Cloud Transformation

Foundational Services

CraigKeenan.Info: Lesson and Lab Guide

Lesson Primer

Amazon Web Service (AWS)

Amazon Web Services Overview

Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Identity Access Management (IAM)

AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources

Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable computing capacity—literally, servers in Amazon's data centers—that you use to build and host your software systems.

Relational Database Service (RDS)

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Virtual Private Cloud (VPC)

What is a Virtual Private Cloud (VPC)

A **virtual private cloud (VPC)** is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings.

VPC and Subnet Basics

A **subnet** is a range of IP addresses in your VPC. You can launch AWS resources into a subnet that you select. Use a public subnet for resources that must be connected to the Internet, and a private subnet for resources that won't be connected to the Internet. For more information about public and private subnets, see VPC and Subnet Basics.

VPC Route Tables

A **route table** contains a set of rules, called *routes*, that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

VPC Internet Gateway

In a VPC, which is otherwise private, instances to be reached from the Internet through an **Internet gateway**.

Security Groups in a VPC

The **security group** that you set up and associate with the instance allows traffic only through specific ports, locking down communication with the instance according to the rules that you specify.

Identity & Access Management (IAM)

IAM Security Credentials

IAM Users can be given Programmatic and/or AWS Management Console Access
Security Credentials can be used for Programmatic Access

IAM Roles

You can use **roles** to delegate access to users, applications, or services that don't normally have access to your AWS resources.

Elastic Compute Cloud (EC2)

Configuration and Credential Files

The CLI stores **credentials** specified with `aws configure` in a local file named `credentials` in a folder named `.aws` in your home directory.

EC2 Instance Metadata

Instance metadata is data about your instance that you can use to configure or manage the running instance.

Amazon Web Services Command Line Interface (AWS CLI)

What is Amazon Web Services Command Line Interface (AWS CLI)?

The **AWS Command Line Interface (CLI)** is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

Relational Database Service (RDS)

Working with DB Subnet Groups

Subnets are segments of a VPC's IP address range that you designate to group your resources based on security and operational needs. A **DB subnet group** is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances.

WordPress

What is WordPress?

Create a free website or easily build a blog on WordPress Hundreds of free, customizable, mobile-ready designs and themes.

WordPress is an online, open source website creation tool written in PHP. But in non-geek speak, it's probably the easiest and most powerful blogging and website content management system (or CMS) in existence today.

Before you begin

Lesson Perquisites

You must have completed all previous lesson(s) before starting on this course.
Understand that Amazon Free Tier Pricing, so don't leave services running unnecessarily.

You only Pay for What You Use

With Amazon Web Services, you only pay for what you use. Since we will be using Free Tier eligible services for most parts of the lecture, our costs should be minimal. Each lesson is design to have you create, modify, and then delete any resources we create. As a matter of best practice, delete any resources you have created when you are done with the lesson. If you intend to pause, and then resume hours or even days later… STOP! Go back and delete any resources you created during this lab to avoid unnecessary charges.

Note:
We will eventually build out a 'Production' like environment with the intention of running our Website. When necessary, we will identify through the procedures what resources to leave running. If not specifically stated to leave the resource, it should be deleted upon completion of the lesson.

Standards and Naming Conventions

Though out these labs, we will provide examples for input/values that must be provided. These examples, (Ex. Example) will be inside parenthesis and highlighted in grey. These are provided to guide you with meaningful values for a lab scenario. On occasion the vendor's documents will provide their own example names. These names will typically be something like *MyExample.* While AWS and our names acceptable for a lab, they are poorly designed for any production or large environment. Try and think of meaningful name for your resources.

Hint: Use your Company Name, domain name, region, environment type, application, etc., or any combination of meaningful data that clearly identifies your resources so support staff doesn't waste time mapping the environment layout.

AWS offers **Tags** for most if not all resources. Tags are meta-data, or data about your data. They help to identify a resources owner, purpose, environment, etc. Tags are **Key/Value** pairs.
Examples of Tags are:

Key	Value
Domain	example.com
Location	us-east-1
Environment	Dev
Application	WordPress
App Owner	Web Admins
Cost Center	2017

IMPORTANT: ALWAYS Use Tags!

Create meaningful tags to help you identify owner, resource purpose, or describe environment.

Lab Overview

Throughout all lessons, we will be acting as if we are an individual who is looking to get started with Amazon Web Services Cloud Computing. Fortunately, we have a friend who is a Cloud Guru, and this friend has agreed to guide us through the process. Our goal is to build a web site for our product/service. It can be anything we want it to be. For teaching purposes, we'll call our experiment OurAmazonWebServices (**OAWS**).

Note: When you see an example using **oaws** or **ouraws** in the name, you must replace with a name unique to your environment.

We're completing the foundations for our website, and will have many of the core components required built by the end of this lab. We need to familiarize ourselves with the core services like VPC, IAM, EC2, and RDS. The goal of the lesson is to:

- Deploy a VPC (Subnets, Internet Gateway, Route Table, and Security Groups)
- Create/use both IAM Security Credentials and IAM Roles (Identify which is recommended)
- Deploy EC2 Instance in VPC Public Subnet
- Install/use the AWS CLI
- Deploy RDS DB in VPC Private Subnet
- Install, Configure and Post to WordPress Web Server

Again, we are working with a minimal budget so everything we do at this time will be in the Free Tier. Pay attention and make sure you select the correct Free Tier services.

Foundational Services - Lab Overview

Virtual Private Cloud (VPC)

- ☐ Create an VPC
- ☐ Adding a Subnet to Your VPC
- ☐ Modifying the Public IPv4 Addressing Attribute of Your Subnet
- ☐ Creating an Internet Gateway and attach it to your VPC
- ☐ Creating a custom Route Table
- ☐ Create Security Groups
- ☐ Add/Update Rules in Security Group

Identity & Access Management (IAM)

- ☐ Creating, Modifying and Viewing Access Keys (AWS Management Console)
- ☐ Creating a Role to Delegate Permissions to an AWS Service

Elastic Compute Cloud (EC2)

- ☐ Launch EC2 Instance in a VPC
- ☐ Configuring the AWS Command Line Interface – Security Credentials
- ☐ Launch EC Instance in a VPC using IAM Role
- ☐ Configuring the AWS Command Line Interface – IAM Role
- ☐ Using EC2 Instance Metadata

Amazon Web Services Command Line Interface (AWS CLI)

- ☐ Using Amazon S3 with AWS Command Line Interface
- ☐ Listing Buckets
- ☐ Creating Buckets
- ☐ Removing Buckets
- ☐ Copy Files into Bucket
- ☐ Update S3 Bucket HTML

Relational Database Service (RDS)

- ☐ Create a DB Subnet Group
- ☐ Creating a DB Instance Running the MySQL Database Engine (in a VPC Private Subnet)

Install WordPress Web Server

- ☐ Update Amazon Linux EC2 Instance Operating System
- ☐ Install httpd, php, and php-mysql
- ☐ Configure HTML HealthCheck
- ☐ Start Apache Web Server (httpd). Enable Auto Start.
- ☐ WordPress Famous 5-Minute Install
- ☐ Restart Apache web server (httpd) service
- ☐ Write WordPress Post
- ☐ Update WordPress Theme

Cleanup!

- ☐ Delete/Stop All Billable Resources
- ☐ Deleting an RDS DB Subnet Group
- ☐ Deleting a Role
- ☐ Deleting a IAM User Security Credentials

- ☐ Deleting a Security Group
- ☐ Detaching an Internet Gateway
- ☐ Deleting an Internet Gateway
- ☐ Deleting Your Subnet
- ☐ Deleting Your VPC
- ☐ Review Environment Configuration

Lab – Foundational Services

Virtual Private Cloud (VPC)

Create a VPC

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#Create-VPC

You can create a VPC and subnets using the Amazon VPC console. The following procedures are for manually creating a VPC and subnets. You also have to manually add gateways and routing tables. Alternatively, you can use the Amazon VPC wizard to create a VPC plus its subnets, gateways, and routing tables in one step.

You can create an empty VPC using the Amazon VPC console.

To create a VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Your VPCs, Create VPC**.
3. Specify the following VPC details as necessary and choose **Create VPC**.
   - Name tag: Optionally provide a name for your VPC. Doing so creates a tag with a key of Name and the value that you specify.
     (Ex. oawsVPC)
   - IPv4 CIDR block: Specify an IPv4 CIDR block for the VPC. We recommend that you specify a CIDR block from the private (non-publicly routable) IP address ranges as specified in RFC 1918;
     (Ex. 10.0.0.0/16)

Adding a Subnet to Your VPC

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#Create-VPC

When you add a new subnet to your VPC, you can specify the Availability Zone in which you want the subnet to reside. You can have multiple subnets in the same Availability Zone. You must specify an IPv4 CIDR block for the subnet from the range of your VPC.

To add a subnet to your VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Subnets**, **Create Subnet**.
3. Specify the subnet details as necessary and choose **Create Subnet**.
   - Name tag: Optionally provide a name for your subnet. Doing so creates a tag with a key of Name and the value that you specify.
     (Ex. oawsPublic-subnet-us-east-1a-10.0.1.0/24)
   - VPC: Choose the VPC for which you're creating the subnet.
     (Ex. oawsVPC)
   - Availability Zone: Optionally choose an Availability Zone in which your subnet will reside, or leave the default No Preference to let AWS choose an Availability Zone for you.
     (Ex. us-east-1a)
   - IPv4 CIDR block: Specify an IPv4 CIDR block for your subnet.
     (Ex. 10.0.1.0/24)
4. (Optional) If required, repeat the steps above to create more subnets in your VPC.
   (Ex. Create two additional subnets:

Tag	VPC	Availability Zone	CIDR
oawsPrivate- subnet-us-east-1a-10.0.2.0/24	oawsVPC	us-east-1a	10.0.2.0./24)
oawsPrivate-subnet-us-east-1b-10.0.3.0/24	oawsVPC	us-east-1b	10.0.3.0./24)

   HINT First subnet will be Public. Remaining two subnets will private subnets.

After you've created a subnet, you can do the following:

- Configure your routing. To make your subnet a public subnet, you must first attach an Internet gateway to your VPC.
- You can then create a custom route table, and add route to the Internet gateway.
- Modify the subnet settings to specify that all instances launched in that subnet receive a public IPv4 address, or an IPv6 address, or both.
- Create or modify your security groups as needed.
- Create or modify your network ACLs as needed.

Modifying the Public IPv4 Addressing Attribute for Your Subnet

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-ip-addressing.html#subnet-public-ip

By default, nondefault subnets have the IPv4 public addressing attribute set to *false*, and default subnets have this attribute set to *true*. An exception is a nondefault subnet created by the Amazon EC2 launch instance wizard — the wizard sets the attribute to *true*. You can modify this attribute using the Amazon VPC console.

To modify your subnet's public IPv4 addressing behavior

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Subnets**.
3. Select your subnet and choose **Subnet Actions**, **Modify auto-assign IP settings**.
4. The **Enable auto-assign public IPv4 address** check box, if selected, requests a public IPv4 address for all instances launched into the selected subnet. **Select** or clear the check box as required, and then choose **Save**.
   HINT Only enable Public IP on your Public (Internet facing) Subnets.

Create Internet Gateway and attach it to your VPC

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html

The following sections describe how to manually create a public subnet to support Internet access.

To create an Internet gateway and attach it to your VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Internet Gateways**, and then choose **Create Internet Gateway**.
3. In the **Create Internet Gateway** dialog box, you can optionally name your Internet gateway, and then choose **Yes, Create**.
   (Ex. oawsIGW)
4. Select the Internet gateway that you just created, and then choose **Attach to VPC**.
5. In the **Attach to VPC** dialog box, select your VPC from the list, and then choose **Yes, Attach**.

Creating a Custom Route Table

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html

When you create a subnet, we automatically associate it with the main route table for the VPC. By default, the main route table doesn't contain a route to an Internet gateway. The following procedure creates a custom route table with a route that sends traffic destined outside the VPC to the Internet gateway, and then associates it with your subnet.

To create a custom route table

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. In the navigation pane, choose **Route Tables**, and then choose **Create Route Table**.
3. In the **Create Route Table** dialog box, optionally name your route table, then select your VPC, and then choose **Yes, Create**.
   (Ex. oawsPublic-RT)
4. Select the custom route table that you just created. The details pane displays tabs for working with its routes, associations, and route propagation.
5. On the **Routes** tab, choose **Edit, Add another route**, and add the following routes as necessary. Choose **Save** when you're done.
   - For IPv4 traffic specify 0.0.0.0/0 in the Destination box, and select the Internet gateway ID in the Target list.
   - For IPv6 traffic, specify ::/0 in the Destination box, and select the Internet gateway ID in the Target list.
6. On the **Subnet Associations** tab, choose **Edit**, select the **Associate** check box for the subnet, and then choose **Save**.
   (Ex. cpPublic-subnet-us-east-1a-10.0.1.0/24)
7. (Optional) If required, repeat the steps above to create a Private Route Table in your VPC.

Name tag	VPC	Destination	Target
oawsPrivate-RT	oawsVPC	10.0.0.0/16	local

Subnet Associations
cpPrivate-subnet-us-east-1a-10.0.2.0/24
cpPrivate-subnet-us-east-1b-10.0.3.0/24
HINT Do not add route to Internet Gateway for Private RDS Subnets.

Creating a Security Group

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html

Although you can use the default security group for your instances, you might want to create your own groups to reflect the different roles that instances play in your system.

To create a security group

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. Enter a name of the security group (for example, `my-security-group`) and provide a description. Select the ID of your VPC from the **VPC** menu and choose **Yes, Create**.
   (Ex. oawsPublic-SG, Description "OAWS Public Security Group")
5. (Optional) If required, create RDS Security Group and add rules for MySQL
   (Ex.

Name tag	Group name	Description	VPC
oawsRDS-SG	oawsRDS-SG	OAWS RDS Security Group	oawsVPC

)

   By default, new security groups start with only an outbound rule that allows all traffic to leave the instances. You must add rules to enable any inbound traffic or to restrict the outbound traffic.

Adding and Removing Rules (to Security Group)

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html

When you add or remove a rule, any instances already assigned to the security group are subject to the change.

To add a rule

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to update. The details pane displays the details for the security group, plus tabs for working with its inbound rules and outbound rules.
4. On the **Inbound Rules** tab, choose **Edit**. Select an option for a rule for inbound traffic for **Type**, and then fill in the required information. For example, for a public web server, choose **HTTP** or **HTTPS** and specify a value for **Source** as `0.0.0.0/0`. Choose **Save**.

   (Ex.

   Inbound Rule

Type	Protocol	Port Range	Source
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
SSH	TCP	2	0.0.0.0/0
RDP	TCP	3389	0.0.0.0/0

)

   Note If you use `0.0.0.0/0`, you enable all IPv4 addresses to access your instance using HTTP or HTTPS. To restrict access, enter a specific IP address or range of addresses.

   HINT Allowing all IPs to access your Public Web Server using HTTP is the required configuration for our lab. However, allowing the whole world access using RDP or SSH is a bad idea for a production environment.
5. You can also allow communication between all instances associated with this security group. On the **Inbound Rules** tab, choose **All Traffic** from the **Type** list. Start typing the ID of the security group for **Source**; this provides you with a list of security groups. Select the security group from the list and choose **Save**.
6. If you need to, you can use the Outbound Rules tab to add rules for outbound traffic.
7. (Optional) If required, update RDS Security Group and add rules for MySQL

   (Ex.

   Inbound Rule

Type	Protocol	Port Range	Source
MySQL/Aurora(3306)	TCP(6)	3306	10.0.0.0/16

)

To delete a rule

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group to update. The details pane displays the details for the security group, plus tabs for working with its inbound rules and outbound rules.
4. Choose **Edit**, select the role to delete, and then choose **Remove, Save**.

Identity & Access Management (IAM)

Creating, Modifying and Viewing Access Keys (AWS Management Console)

URL Link: http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

Users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI). To fill this need, you can create, modify, view, or rotate access keys (access key IDs and secret access keys) for IAM users.

When you create an access key, IAM returns the access key ID and secret access key. You should save these in a secure location and give them to the user.

Important

To ensure the security of your AWS account, the secret access key is accessible only at the time you create it. If a secret access key is lost, you must delete the access key for the associated user and create a new key.

Prerequisites

- Create IAM User (Console)
  (Ex. oawsAdmin or craigkeenan)

To list a user's access keys

1. Sign in to the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. Choose the name of the desired user, and then choose the **Security Credentials** tab. The user's access keys and the status of each key is displayed.

   Note

   Only the user's access key ID is visible. The secret access key can only be retrieved when creating the key.

To create, modify, or delete a user's access keys

1. Sign in to the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. Choose the name of the desired user, and then choose the **Security Credentials** tab.
4. If needed, expand the Access Keys section and do the following:
   - To create an access key, choose **Create Access Key** and then choose **Download Credentials** to save the access key ID and secret access key to a CSV file on your computer. Store the file in a secure location. You will not have access to the secret access key again after this dialog box closes. After you have downloaded the CSV file, choose **Close**.

Creating a Role to Delegate Permissions to an AWS Service

URL Link: http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-service.html

You create a role for an AWS service when you want to grant permissions to a service like Amazon EC2, AWS S3, or Amazon RDS. These services can access AWS resources, so you create a role to determine what the service is allowed to do with those resources.

To create a role for an AWS service using the AWS Console

1. Sign in to the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the IAM console, click **Roles**, and then click **Create New Role**.
3. For **Role name**, type a role name that can help you identify the purpose of this role. Role names must be unique within your AWS account. After you type the name, click **Next Step** at the bottom of the page.
   (Ex. oawsAdminRole)

Because various entities might reference the role, you cannot change the name of the role after it has been created.

Important Role names must be unique within an account. They are not distinguished by case, for example, you cannot create roles named both "PRODROLE" and "prodrole".

4. Expand the **AWS Service Roles** section, and then select the service that you want to allow to assume this role.
   (Ex. To create for Amazon EC2, Click **Select** next to **Amazon EC2)**
5. Select the check box for a managed policy that grants the permissions that you want the service to have. If the policy does not yet exist, then you can skip this step, create the policy later, and then attach it to the role.
   (Ex. AdministratorAccess)
6. Click **Next Step** to review the role. Then click **Create Role**.

Elastic Compute Cloud (EC2)

Launch EC2 Instance in a VPC

URL Link: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html

An instance is a virtual server in the AWS cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

When you sign up for AWS, you can get started with Amazon EC2 for free using the AWS Free Tier. You can either leverage the free tier to launch and use a micro instance for free for 12 months.

To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation bar at the top of the screen, the current region is displayed. Select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see Resource Locations.
3. From the Amazon EC2 console dashboard, choose **Launch Instance**.
4. On the Choose an **Amazon Machine Image (AMI)** page, choose an AMI and then choose **Select**.
   (Ex. **Amazon Linux AMI 2016.09.1 (HVM), SSD Volume Type** - ami-0b33d91d – Free Tier eligible)
5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory.
   To remain eligible for the free tier, choose the t2.micro instance type.
   Choose **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, change the following settings as necessary (expand Advanced Details to see all the settings), and then choose **Next: Add Storage**:
   (Ex.
   Number of instances: 1
   Network: Select oawsVPC
   Subnet: Select subnet-us-east-1a-10.0.1.0/24
   Auto-assign Public IP: Select Use subnet setting (Enable)
)
   - **Number of instances**: Enter the number of instances to launch.
   - **Purchasing option**: Select Request Spot instances to launch a Spot instance. into EC2-Classic:
   - **Network**: Select the VPC, or to **create a new VPC**.
   - **Subnet**: Select the subnet into which to launch your instance.

- **Auto-assign Public IP**: Specify whether your instance receives a public IPv4 address. By default, instances in a default subnet receive a public IPv4 address and instances in a nondefault subnet do not. You can select Enable or Disable to override the subnet's default setting.
- **Auto-assign IPv6 IP**: Specify whether your instance receives an IPv6 address from the range of the subnet. Select Enable or Disable to override the subnet's default setting. This option is only available if you've associated an IPv6 CIDR block with your VPC and subnet.
- **IAM role**: Select an AWS Identity and Access Management (IAM) role to associate with the instance.
- **Shutdown behavior**: Select whether the instance should stop or terminate when shut down.
- **Enable termination protection**: Select this check box to prevent accidental termination.
- **Monitoring**: Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply.
- **EBS-Optimized instance**: An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, select this check box to enable it. Additional charges apply.
- **Tenancy**: If you are launching your instance into a VPC, you can choose to run your instance on isolated, dedicated hardware (Dedicated) or on a Dedicated host (Dedicated host). Additional charges may apply.
- **Network interfaces**: If you selected a specific subnet, you can specify up to two network interfaces for your instance:
- **Kernel ID**: (Only valid for paravirtual (PV) AMIs) Select Use default unless you want to use a specific kernel.
- **RAM disk ID**: (Only valid for paravirtual (PV) AMIs) Select Use default unless you want to use a specific RAM disk. If you have selected a kernel, you may need to select a specific RAM disk with the drivers to support it.
- **Placement group**: A placement group is a logical grouping for your cluster instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups.

  User data: You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the As file option and browse for the file to attach.

7. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume). You can change the following options, then choose **Next: Add Tags** when you have finished:
8. On the **Add Tags** page, specify tags for the instance by providing key and value combinations. Choose **Add another tag** to add more than one tag to your resource. Choose **Next: Configure Security Group** when you are done.

   (Ex. **Key** = Name ; **Value** = oawsEC2VPC)
9. On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. Select or create a security group as follows, and then choose **Review and Launch**.

   To select an existing security group:

   Choose Select an existing security group. Your security groups are displayed.
   Select a security group from the list.
   (Ex. oawsPublic-SG)
10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link.
    When you are ready, choose **Launch**.
11. 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. For example, choose **Choose an existing key pair**, then select the key pair you created when getting set up.
    To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

Important If you choose the Proceed without key pair option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

Configuring the AWS Command Line Interface – Security Credentials

URL Link: http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html

This section explains how to configure settings that the AWS Command Line Interface uses when interacting with AWS, such as your security credentials and the default region.

HINT AWS CLI will not work because it needs credentials. Lab will install credentials. Issues is that anyone with access to your Security Credentials can add/modify/delete your resources. This is UNSAFE!

Prerequisite

Launch EC2 Instance
SSH to EC2 Instance
Test AWS CLI Functionality (For example, listing S3 Buckets)

```
aws s3 ls
```

Quick Configuration

For general use, the aws configure command is the fastest way to set up your AWS CLI installation.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-east-1
Default output format [None]:
```

The AWS CLI will prompt you for four pieces of information. **AWS Access Key ID** and **AWS Secret Access Key** are your account credentials.

Default region is the name of the region you want to make calls against by default. This is usually the region closest to you, but it can be any region.

Note

You must specify an AWS region when using the AWS CLI. For a list of services and available regions, see Regions and Endpoints.

Default output format can be either json, text, or table. If you don't specify an output format, json will be used.

Configuration and Credential Files

The CLI stores credentials specified with *aws configure* in a local file named *credentials* in a folder named *.aws* in your home directory. Home directory location varies but can be referred to using the environment variables *%UserProfile%* in Windows and *$HOME* or ~ (tilde) in Unix-like systems.

For example, the following commands list the contents of the .aws folder:

Linux, macOS, or Unix

```
$ ls ~/.aws
```

Windows

```
> dir %UserProfile%\.aws
```

In order to separate credentials from less sensitive options, region and output format are stored in a separate file named *config* in the same folder.

The files generated by the CLI for the profile configured in the previous section look like this:

```
more ~/.aws/credentials
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

more ~/.aws/config
[default]
region=us-west-2
output=json
```

The following settings are supported.

aws_access_key_id – AWS access key.
aws_secret_access_key – AWS secret key.
aws_session_token – AWS session token. A session token is only required if you are using temporary security credentials.
region – AWS region.
output – output format (json, text, or table)

Launch EC2 Instance in a VPC using IAM Role

URL Link: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles as follows:

To launch an instance with an IAM role using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation bar at the top of the screen, the current region is displayed. Select the region for the instance. This choice is important because some Amazon EC2 resources can be shared between regions, while others can't. Select the region that meets your needs. For more information, see Resource Locations.
3. From the Amazon EC2 console dashboard, choose **Launch Instance**.
4. On the Choose an **Amazon Machine Image (AMI)** page, choose an AMI and then choose **Select**.
   (Ex. **Amazon Linux AMI 2016.09.1 (HVM), SSD Volume Type** - ami-0b33d91d – Free Tier eligible)
5. On the **Choose an Instance Type** page, select the hardware configuration and size of the instance to launch. Larger instance types have more CPU and memory.
   To remain eligible for the free tier, choose the t2.micro instance type.
   Choose **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page, change the following settings as necessary (expand Advanced Details to see all the settings), and then choose **Next: Add Storage**:
   (Ex.
   Number of instances: 1
   Network: Select oawsVPC
   Subnet: Select subnet-us-east-1a-10.0.1.0/24
   Auto-assign Public IP: Select Use subnet setting (Enable)
   IAM Role: oawsAdminRole
)
   - **Number of instances**: Enter the number of instances to launch.
   - **Purchasing option**: Select Request Spot instances to launch a Spot instance. into EC2-Classic:

- **Network**: Select the VPC, or to **create a new VPC**.
- **Subnet**: Select the subnet into which to launch your instance.
- **Auto-assign Public IP**: Specify whether your instance receives a public IPv4 address. By default, instances in a default subnet receive a public IPv4 address and instances in a nondefault subnet do not. You can select Enable or Disable to override the subnet's default setting.
- **Auto-assign IPv6 IP**: Specify whether your instance receives an IPv6 address from the range of the subnet. Select Enable or Disable to override the subnet's default setting. This option is only available if you've associated an IPv6 CIDR block with your VPC and subnet.
- **IAM role**: Select an AWS Identity and Access Management (IAM) role to associate with the instance.
- **Shutdown behavior**: Select whether the instance should stop or terminate when shut down.
- **Enable termination protection**: Select this check box to prevent accidental termination.
- **Monitoring**: Select this check box to enable detailed monitoring of your instance using Amazon CloudWatch. Additional charges apply.
- **EBS-Optimized instance**: An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. If the instance type supports this feature, select this check box to enable it. Additional charges apply.
- **Tenancy**: If you are launching your instance into a VPC, you can choose to run your instance on isolated, dedicated hardware (Dedicated) or on a Dedicated host (Dedicated host). Additional charges may apply.
- **Network interfaces**: If you selected a specific subnet, you can specify up to two network interfaces for your instance:
- **Kernel ID**: (Only valid for paravirtual (PV) AMIs) Select Use default unless you want to use a specific kernel.
- **RAM disk ID**: (Only valid for paravirtual (PV) AMIs) Select Use default unless you want to use a specific RAM disk. If you have selected a kernel, you may need to select a specific RAM disk with the drivers to support it.
- **Placement group**: A placement group is a logical grouping for your cluster instances. Select an existing placement group, or create a new one. This option is only available if you've selected an instance type that supports placement groups.

User data: You can specify user data to configure an instance during launch, or to run a configuration script. To attach a file, select the As file option and browse for the file to attach.

7. On the **Add Storage** page, you can specify volumes to attach to the instance besides the volumes specified by the AMI (such as the root device volume). You can change the following options, then choose **Next: Add Tags** when you have finished:
8. On the **Add Tags** page, specify tags for the instance by providing key and value combinations. Choose **Add another tag** to add more than one tag to your resource. Choose **Next: Configure Security Group** when you are done.

   (Ex. **Key** = Name ; **Value** = oawsEC2VPC)
9. On the **Configure Security Group** page, use a security group to define firewall rules for your instance. These rules specify which incoming network traffic is delivered to your instance. All other traffic is ignored. Select or create a security group as follows, and then choose **Review and Launch**.

   To select an existing security group:

   Choose Select an existing security group. Your security groups are displayed.
   Select a security group from the list.
   (Ex. oawsPublic-SG)
10. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by choosing the appropriate **Edit** link.
    When you are ready, choose **Launch**.

11. 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. For example, choose **Choose an existing key pair**, then select the key pair you created when getting set up.
    To launch your instance, select the acknowledgment check box, then choose **Launch Instances**.

    Important If you choose the Proceed without key pair option, you won't be able to connect to the instance unless you choose an AMI that is configured to allow users another way to log in.

Configuring the AWS Command Line Interface – IAM Role

URL Link: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

This section explains how to use the AWS Command Line Interface using roles

HINT Using IAM Roles is recommended over storing Security Credentials on EC2 Instance.

Prerequisite

Launch EC2 Instance with IAM Role
SSH to EC2 Instance
Test AWS CLI Functionality (For example, listing S3 Buckets)

```
aws s3 ls
```

Quick Configuration

For general use, the aws configure command is the fastest way to set up your AWS CLI installation.

```
$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: us-east-1
Default output format [None]:
```

We do not need to provide AWS Access Key ID and AWS Secret Access Key for your account credentials when using IAM roles.
Default region is the name of the region you want to make calls against by default. This is usually the region closest to you, but it can be any region.

Note

You must specify an AWS region when using the AWS CLI. For a list of services and available regions, see Regions and Endpoints.

Instance Metadata

URL Link: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html

Instance metadata is data about your instance that you can use to configure or manage the running instance.

Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata.

Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

Prerequisite
Launch EC2 Instance
SSH to EC2 Instance

To view all categories of instance metadata from within a running instance, use the following URI:

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

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.
You can use a tool such as cURL, or if your instance supports it, the GET command; for example:

```
$ curl http://169.254.169.254/latest/meta-data/
$ GET http://169.254.169.254/latest/meta-data/
$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-12345678)
```

Amazon Web Services Command Line Interface (CLI)

Using Amazon S3 with the AWS Command Line Interface

URL Link: http://docs.aws.amazon.com/cli/latest/userguide/cli-s3.html

The AWS CLI provides two tiers of commands for accessing Amazon S3.

- The first tier, named s3, consists of high-level commands for frequently used operations, such as creating, manipulating, and deleting objects and buckets.
- The second tier, named s3api, exposes all Amazon S3 operations, including modifying a bucket access control list (ACL), using cross-origin resource sharing (CORS), or logging policies. It allows you to carry out advanced operations that may not be possible with the high-level commands alone.

Prerequisite

- Launch EC2 Instance with IAM Role
- SSH to EC2 Instance
- AWS CLI Installed on Instance (By default AWS CLI is installed on Amazon Linux AMIs)

To get a list of all commands available in each tier, use the help argument with the aws s3 or aws s3api commands:

```
$ aws s3 help
```

or

```
$ aws s3api help
```

Note
The AWS CLI supports copying, moving, and syncing from Amazon S3 to Amazon S3. These operations use the service-side COPY operation provided by Amazon S3: Your files are kept in the cloud, and are not downloaded to the client machine, then back up to Amazon S3.
When operations such as these can be performed completely in the cloud, only the bandwidth necessary for the HTTP request and response is used.

Listing Buckets

To list all buckets or their contents, use the aws s3 ls command. Here are some examples of common usage.
The following command lists all buckets.

```
$ aws s3 ls
      CreationTime Bucket
      ------------ ------
2013-07-11 17:08:50 my-bucket
```

```
2013-07-24 14:55:44 my-bucket2
```

The following command lists all objects and folders (prefixes) in a bucket.

```
$ aws s3 ls s3://bucket-name
Bucket: my-bucket
Prefix:

      LastWriteTime     Length Name
      -------------     ------ ----
                           PRE path/
2013-09-04 19:05:48          3 MyFile1.txt
```

The following command lists the objects in bucket-name/path (in other words, objects in bucket-name filtered by the prefix path).

```
$ aws s3 ls s3://bucket-name/path
Bucket: my-bucket
Prefix: path/

      LastWriteTime     Length Name
      -------------     ------ ----
2013-09-06 18:59:32          3 MyFile2.txt
```

Creating Buckets

Use the *aws s3 mb* command to create a new bucket. Bucket names must be unique and should be DNS compliant. Bucket names can contain lowercase letters, numbers, hyphens and periods. Bucket names can only start and end with a letter or number, and cannot contain a period next to a hyphen or another period.

```
$ aws s3 mb s3://bucket-name
```

Removing Buckets

To remove a bucket, use the *aws s3 rb* command.

```
$ aws s3 rb s3://bucket-name
```

By default, the bucket must be empty for the operation to succeed. To remove a non-empty bucket, you need to include the --force option.

```
$ aws s3 rb s3://bucket-name --force
```

This will first delete all objects and subfolders in the bucket and then remove the bucket.

Note

If you are using a versioned bucket that contains previously deleted—but retained—objects, this command will not allow you to remove the bucket.

Copy Files into Bucket

The following example copies an object into a bucket.

```
aws s3 cp file.txt s3://my-bucket/
```

Update S3 Bucket HTML

Follow these procedures to test a few new HTML sample pages for your site

1. Launch EC2 Instance with IAM Role
2. SSH into EC2 Instance
3. Elevate to Root Privlege

   ```
   sudo su
   ```
4. Copy sample HTML into S3 Bucket

```
aws s3 cp –recursive s3://www.ouraws.com/code/HTML/00/ s3://my-bucket/
```

5. View S3 Static Website to see updated HTML
   (Ex. Locate S3 Static Website Hosting Endpoint in S3 Bucket Properties – Click Link)
6. Repeat Steps 4 & 5 with following:

```
aws s3 cp –recursive s3:// www.ouraws.com/code/HTML/01/ s3://my-bucket/
aws s3 cp –recursive s3:// www.ouraws.com/code/HTML/02/ s3://my-bucket/
aws s3 cp –recursive s3:// www.ouraws.com/code/HTML/03/ s3://my-bucket/
```

Relational Database Service (RDS)

Create a DB Subnet Group

URL Link:
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.WorkingWithRDSInstanceinaVPC.html#USER_VPC.CreateDBSubnetGroup

A DB subnet group is a collection of subnets (typically private) that you create for a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when you create DB instances. Each DB subnet group must have at least one subnet in at least two Availability Zones in the region.

Note
When you create a DB instance in a VPC, you must select a DB subnet group. Amazon RDS then uses that DB subnet group and your preferred Availability Zone to select a subnet and an IP address within that subnet.

In this step, you create a DB subnet group and add the subnets you created for your VPC.

AWS Management Console

To create a DB subnet group

1. Open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation pane, choose **Subnet Groups**.
3. Choose **Create DB Subnet Group**.
4. For **Name**, type the name of your DB subnet group.
   (Ex. oawsRDS-DBSubnetGroup)
5. For **Description**, type a description for your DB subnet group.
   (Ex. oawsRDS-DBSubnetGroup)
6. For **VPC ID**, choose the VPC that you created.
   (Ex. oawsVPC)
7. In the **Add Subnet(s) to this Subnet Group** section, click the **add all the subnets link** or Choose **Availability Zone** and **Subnet ID** and click **Add** to add individual subnets.
   (Ex.
   oawsPrivate-subnet-us-east-1a-10.0.2.0/24
   oawsPrivate-subnet-us-east-1b-10.0.3.0/24
)
8. Choose **Yes, Create**, and then choose **Close**.

Your new DB subnet group appears in the DB subnet groups list on the RDS console. You can click the DB subnet group to see details, including all of the subnets associated with the group, in the details pane at the bottom of the window.

Creating a DB Instance Running the MySQL Database Engine (in a VPC Private Subnet)

URL Link: http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateInstance.html

The basic building block of Amazon RDS is the DB instance. The DB instance is where you create your MySQL databases.

AWS Management Console
To launch a MySQL DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the top right corner of the AWS Management Console, select the region in which you want to create the DB instance.
3. In the navigation pane, click **Instances**.
4. Click **Launch DB Instance** to start the **Launch DB Instance Wizard**. The wizard opens on the **Select Engine** page
   (Ex. Choose MySQL which has free tier offering).
5. In the **Launch DB Instance Wizard** window, click the **Select** button for the MySQL DB engine.
6. The next step asks if you are planning to use the DB instance you are creating for production. If you are, select **Yes**. By selecting **Yes**, the failover option **Multi-AZ** and the **Provisioned IOPS** storage option will be preselected in the following step. Click **Next** when you are finished.
   (Ex. Select **Dev/Test MySQL**)
7. On the **Specify DB Details** page, specify your DB instance information. The following table shows settings for an example DB instance. Click **Next** when you are finished.
   (Ex.
   DB Instance Class: **db.t2.micro**
   Multi-AZ Deployment: **No**
   DB Instance Identifier: **oawsDB**
   Master Username: **dbadmin**
   Master Password: **dbadmin!**
   Confirm Password: **dbadmin!**
)

For this parameter...	...Do this:
License Model	MySQL has only one license model. Select the default, **`General-Public-License`**, to use the general license agreement for MySQL.
DB Engine Version	Select the version of MySQL that you want to work with. Note that Amazon RDS supports several versions of MySQL.
DB Instance Class	Select a DB instance class that defines the processing and memory requirements for the DB instance. For more information about all the DB instance class options, see DB Instance Class.
Multi-AZ Deployment	Determine if you want to create a standby replica of your DB instance in another Availability Zone for failover support. For more information about multiple Availability Zones, see Regions and Availability Zones.
Allocated Storage	Type a value to allocate storage for your database (in gigabytes). In some cases, allocating a higher amount of storage for your DB instance than the size of your database can improve I/O performance. For more information about storage allocation, see Amazon RDS Storage Types.
Storage Type	Select the storage type you want to use. For more information about storage, see Storage for Amazon RDS.

DB Instance Identifier	Type a name for the DB instance that is unique for your account in the region you selected. You may choose to add some intelligence to the name such as including the region and DB Engine you selected, for example **`mysql-instance1`**.
Master Username	Type a name using alphanumeric characters that you will use as the master user name to log on to your DB instance. The default privileges granted to the master user name account include: create, drop, references, event, alter, delete, index, insert, select, update, create temporary tables, lock tables, trigger, create view, show view, alter routine, create routine, execute, create user, process, show databases, grant option.
Master Password	Type a password that contains from 8 to 16 printable ASCII characters (excluding /,", and @) for your master user password.
Confirm Password	Re-type the Master Password for confirmation.

8. On the **Configure Advanced Settings** page, provide additional information that RDS needs to launch the MySQL DB instance. The table shows settings for an example DB instance. Specify your DB instance information, then click **Next Step**.

   (Ex.
   VPC: oawsVPC
   DB Subnet Group: oawsrds-dbsubnetgroup
   Publicly Accessible: No
   Availability Zone: us-east-1a
   DB Security Groups: oawsRDS-SG
   Database Name: oawsDB
   Backup Retention Period: 1 day
)
   HINT Set Backup Retention Period to 1 day to avoid storage charges. (10GB/month Free Tier)

For this parameter...	...Do this:
VPC	Select the name of the Virtual Private Cloud (VPC) that will host your MySQL DB instance. If your DB instance will not be hosted in a VPC, select **Not in VPC**. For more information about VPC, see Virtual Private Clouds (VPCs) and Amazon RDS.
DB Subnet Group	This setting depends on the platform you are on. If you are a new customer to AWS, select **`default`**, which will be the default DB subnet group that was created for your account. If you are creating a DB instance on the previous E2-Classic platform and you want your DB instance in a specific VPC, select the DB subnet group you created for that VPC. For more information about VPC, see Virtual Private Clouds (VPCs) and Amazon RDS.
Publicly Accessible	Choose **`Yes`** to give the DB instance a public IP address, meaning that it will be accessible outside the VPC (the DB instance also needs to be in a public subnet in the VPC); otherwise, choose **`No`**, so the DB instance will only be accessible from inside the VPC. For more information about hiding DB instances from public access, see Hiding a DB Instance in a VPC from the Internet.
Availability Zone	Determine if you want to specify a particular Availability Zone. If you selected **Yes** for the Multi-AZ Deployment parameter on the previous page, you will not have any options here. For more information about Availability Zones, see Regions and Availability Zones.
DB Security Groups	Select the security group you want to use with this DB instance. For more information about security groups, see Working with DB Security Groups.

Database Name	Type a name for your default database of 1 to 64 alpha-numeric characters. If you do not provide a name, Amazon RDS will not create a database on the DB instance you are creating. To create additional databases, connect to the DB instance and use the SQL command CREATE DATABASE. For more information about connecting to the DB instance, see Connecting to a DB Instance Running the MySQL Database Engine.
Database Port	Specify the port that applications and utilities will use to access the database. MySQL installations default to port 3306. The firewalls at some companies block connections to the default MySQL port. If your company firewall blocks the default port, choose another port for the new DB instance.
DB Parameter Group	Select a **DB Parameter Group**, which is used to manage your DB engine configuration. Each MySQL version has a default parameter group you can use, or you can create your own parameter group. For DB engine configurations that you frequently use or to increase your database engine uptime, you can create your own DB parameter group. For more information about parameter groups, see Working with DB Parameter Groups.
Option Group	Select an **Option Group**, which is used to enable and configure DB engine features. Each MySQL version has a default option group you can use, or you can create your own option group. For more information about option groups, see Working with Option Groups.
Copy Tags To Snapshots	Choose this option to have any DB instance tags copied to a DB snapshot when you create a snapshot. For more information, see Tagging Amazon RDS Resources.
Enable Encryption	Select Yes to enable encryption at rest for this DB instance. For more information, see Encrypting Amazon RDS Resources.
Backup Retention Period	Select the number of days for Amazon RDS to automatically back up your DB instance. You can recover your database to any point in time during that retention period. For more information, see Working With Backups.
Backup Window	Specify the period of time during which your DB instance is backed up. During the backup window, storage I/O may be suspended while your data is being backed up and you may experience elevated latency. This I/O suspension typically lasts for the duration of the snapshot. This period of I/O suspension is shorter for Multi-AZ DB deployments, since the backup is taken from the standby, but latency can occur during the backup process. For more information, see Working With Backups.
Enable Enhanced Monitoring	Choose **Yes** to enable gathering metrics in real time for the operating system that your DB instance runs on. For more information, see Enhanced Monitoring.
Granularity	Only available if **Enable Enhanced Monitoring** is set to **Yes**. Set the interval, in seconds, between when metrics are collected for your DB instance.
Auto Minor Version Upgrade	Select **Yes** if you want to enable your DB instance to receive minor DB Engine version upgrades automatically when they become available.
Maintenance Window	Select the weekly time range during which system maintenance can occur. For more information about the maintenance window, see Adjusting the Preferred DB Instance Maintenance Window.

In addition, Federated Storage Engine is currently not supported by Amazon RDS for MySQL.

(Ex. Database Name : **oawsDB)**

9. Click **Launch DB Instance** to create your MySQL DB instance.
10. On the final page of the wizard, click **Close**.
11. On the Amazon RDS console, the new DB instance appears in the list of DB instances. The DB instance will have a status of **creating** until the DB instance is created and ready for use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and store allocated, it could take several minutes for the new instance to be available.

Install and Configure WordPress Web Server

Update Amazon Linux EC2 Instance Operating System

URL Link: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-updates.html

To update all packages on an Amazon Linux instance

```
sudo su
yum update –y
```

Install httpd, php, and php-mysql

URL Link: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-updates.html

To install selected packages on Amazon Linux Instance

```
sudo yum install httpd php php-mysql –y
sudo yum install php-gd -y
```

Start Apache web server (httpd) service. Enable Auto-Start

URL Link: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html

To start Apache web server

```
sudo service httpd start
```

To start Apache web server at system boot

```
sudo chkconfig httpd on
```

WordPress Famous 5-Minute Install

URL Link: https://codex.wordpress.org/Installing_WordPress#Famous_5-Minute_Install

WordPress is well-known for its ease of installation. Under most circumstances, installing WordPress is a very simple process and takes less than five minutes to complete. WordPress is used on 80% of all Websites.

Prerequisite

- Launch EC2 Instance with IAM Role
- SSH to EC2 Instance
- httpd, php and php-mysql installed

Here's the quick version of the instructions for those who are already comfortable with performing such installations.

If you are not comfortable with renaming files, step 3 is optional and you can skip it as the install program will create the wp-config.php file for you.

1. Download and unzip the WordPress package if you haven't already.

   ```
   sudo su
   cd /var/www/html
   wget https://wordpress.org/latest.tar.gz
   ```

2. Create a database for WordPress on your web server, as well as a MySQL (or MariaDB) user who has all privileges for accessing and modifying it.

   (Ex. oawsDB)

3. (Optional) Find and rename wp-config-sample.php to wp-config.php, then edit the file (see Editing wp-config.php) and add your database information.
4. Upload the WordPress files to the desired location on your web server:

```
sudo su
cd /var/www/html
tar -xzf latest.tar.gz
cp -r wordpress/* /var/www/html/
rm -rf wordpress
rm -rf latest.tar.gz
chmod -R 755 wp-content
chown -R apache.apache wp-content
```

5. Run the WordPress installation script by accessing the URL in a web browser. This should be the URL where you uploaded the WordPress files.
   (Ex. <Public IP Address of EC2 Instance>/)
   HINT Copy Installer output. Create /var/www/html/wp-config.php file, paste installer output.
6. If you installed WordPress in the root directory, you should visit: http://example.com/

That's it! WordPress should now be installed.

HINT Relaunch web page. Complete Site Setup!

Restart Apache web server (httpd) service.

URL Link: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html

To start Apache web server

```
sudo service httpd restart
```

Write WordPress Post

URL Link: https://codex.wordpress.org/Writing_Posts

To write a post:

1. Log in to your WordPress Administration Panel (Dashboard).
2. Click the '**Posts**' tab.
3. Click the '**Add New**' sub-tab.
4. Start filling in the blanks: enter your post title in the upper field, and enter your post body content in the main post editing box below it.
5. As needed, select a category, add tags, and make other selections from the sections below the post. (Each of these sections is explained below.)
6. When you are ready, click **Publish**.

Update WordPress Theme

URL Link: https://developer.wordpress.org/themes/getting-started/what-is-a-theme/

A WordPress theme changes the design of your website, often including its layout. Changing your theme changes how your site looks on the front-end, i.e. what a visitor sees when they browse to your site on the web. There are thousands of free WordPress themes in the WordPress.org Theme Directory, though many WordPress sites use custom themes.

Activate a Theme

To activate a theme, go to *My Sites* → *Themes* in any site's dashboard or the Theme Showcase. Click on the three dots to the right of the theme's name and then click on **Activate**.

HINT You may need to manually download the themes. Simple Storage Service (S3) is a great place to store the theme so you can copy it (aws s3 cp) to the EC2 /var/www/html/wordpress/themes directory.

Cleanup!

Delete / Stop All Billable Resources

Note: You only pay for what you use. **Why pay if you are not using it?** Shut it down. Clean up. You'll build what you need in the next lab.

To clean up after lesson
Terminate All EC2 Instances
Delete Your MySQL DB Instance
Delete Your DB Subnet Group
Delete IAM Security Credentials
Delete IAM Roles
Detach Your Internet Gateway
Delete Your Internet Gateway
Delete Your Subnets
Delete Your VPC

HINT While we recommend you practice creating and deleting resources, we will need some resources for future lessons. Practice deleting, then rebuild to the required configuration before next lesson.

Deleting an RDS DB Subnet Group

URL Link: http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.html

To delete a DB subnet group

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation pane, click **Subnet Groups**.
3. On the **Subnet Groups** tab, in the **DB subnet group** list, click the row of the group you want to delete.
   (Ex. oawsrds-dbsubnetgroup)
4. In the **Delete DB Subnet Group** dialog box, click **Delete**.
   HINT You will probably need to delete your DB Instance first before executing this procedure

Deleting a Role (AWS Management Console)

URL Link: http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_manage_delete.html

To use the AWS Management Console to delete a role

1. Sign in to the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**, and then select the check box next to the role name that you want to delete, not the name or row itself.
3. For **Role Actions** at the top of the page, choose **Delete Role**.
4. In the confirmation dialog box, review the service last accessed data, which shows when each of the selected roles last accessed an AWS service. This helps you to confirm whether the role is currently active. If you want to proceed, choose **Yes, Delete**. If you are sure, you can proceed with the deletion even if the service last accessed data is still loading.

Deleting a IAM User Security Credentials (AWS Management Console)

URL Link: http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

To delete a user's access keys

1. Sign in to the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. Choose the name of the desired user, and then choose the **Security Credentials** tab.
4. If needed, expand the Access Keys section and do any of the following:
   - To delete an access key, choose **Delete** and then choose **Delete** to confirm.

Deleting a Security Group

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html

You can delete a security group only if there are no instances assigned to it (either running or stopped). You can assign the instances to another security group before you delete the security group (see Changing an Instance's Security Groups). You can't delete a default security group.

To delete a security group

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Select the security group and choose **Security Group Actions**, **Delete Security Group**.
4. In the Delete Security Group dialog box, choose **Yes, Delete**.

Detaching an Internet Gateway from Your VPC

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html

If you no longer need Internet access for instances that you launch into a nondefault VPC, you can detach an Internet gateway from a VPC. You can't detach an Internet gateway if the VPC has instances with associated Elastic IP addresses.

To detach an Internet gateway

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Internet Gateways**.
3. Select the Internet gateway and choose **Detach from VPC**.
4. In the **Detach from VPC** dialog box, choose **Yes, Detach**.

Deleting an Internet Gateway

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html

If you no longer need an Internet gateway, you can delete it. You can't delete an Internet gateway if it's still attached to a VPC.

To delete an Internet gateway

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Internet Gateways**.
3. Select the Internet gateway and choose **Delete**.
4. In the **Delete Internet Gateway** dialog box, choose **Yes, Delete**.

Deleting Your Subnet

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#Create-VPC

If you no longer need your subnet, you can delete it. You must terminate any instances in the subnet first.

To delete your subnet

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Terminate all instances in the subnet.
3. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
4. In the navigation pane, choose **Subnets**.
5. Select the subnet to delete and choose **Subnet Actions**, **Delete**.
6. In the **Delete Subnet** dialog box, choose **Yes, Delete**.

Deleting Your VPC

URL Link: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#Create-VPC

You can delete your VPC at any time. However, you must terminate all instances in the VPC first. When you delete a VPC using the Amazon VPC console, we delete all its components, such as subnets, security groups, network ACLs, Internet gateways, VPC peering connections, and DHCP options.

To delete your VPC

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Terminate all instances in the VPC. For more information, see Terminate Your Instance in the Amazon EC2 User Guide for Linux Instances.
3. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
4. In the navigation pane, choose **Your VPCs**.
5. Select the VPC to delete and choose **Actions**, **Delete VPC**.
6. To delete the VPN connection, select the option to do so; otherwise, leave it unselected. Choose **Yes, Delete**.

Environment Configuration

At the end of this lab, your environment should be configured as follows

Lesson 3

Virtual Private Cloud (VPC)

1. Custom VPC Created (1 VPC)
2. VPC Subnets Created (1 Public Subnet, 2 Private Subnets)
3. Public Subnet Auto-Assign Public IPv4 Enabled
4. Internet Gateway Created and Attached to VPC
5. Public Route Table Created and Associated to Public Subnet
6. Private Route Table Created and Associated to Private Subnets
7. Public Security Created with Inbound Rules for HTTP, HTTPS, SSH an RDP added

Identity and Access Management (IAM)

8. Admin Role Created

Elastic Compute Cloud (EC2)

9. EC2 Instance Created in VPC Public Subnet
10. AWS Command Line Interface (CLI)
11. AWS CLI Installed on EC2 Instance using Roles.

Simple Storage Service (S3)

12. HTML Samples Copied to S3 Bucket

Relational Database Services (RDS)

13. DB Subnet Group Created
14. MySQL DB Instance in VPC Private Subnets
15. MySQL DB Instance Backup Retention Period set to 1 day

Knowledge Base

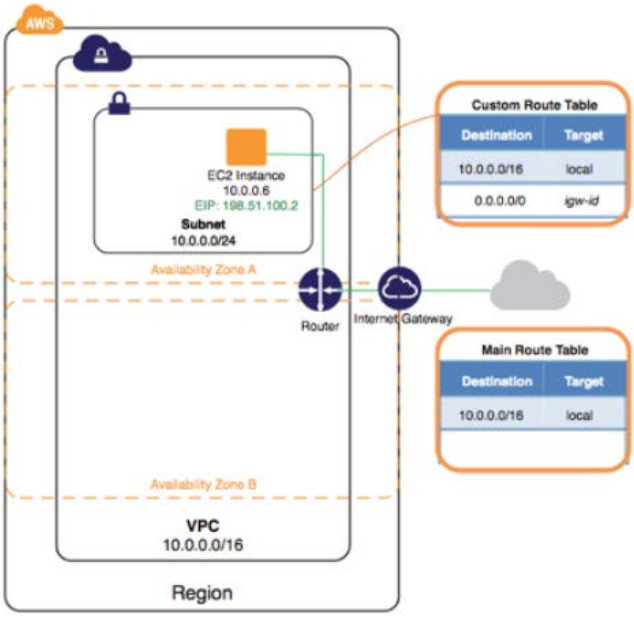
Virtual Private Cloud (VPC)

VPC Overview

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/GettingStartedGuide/ExerciseOverview.html

A virtual private cloud (VPC) is a virtual network that closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of Amazon Web Services (AWS). After you complete the tasks in this exercise, you'll have an Amazon EC2 instance running in a VPC that you can access from the Internet using SSH (for Linux instances) or Remote Desktop (for Windows instances).
For an overview of Amazon VPC, see What is Amazon VPC? in the Amazon VPC User Guide.

The following diagram shows the architecture that you'll create as you complete the exercise in this guide. The security group that you set up and associate with the instance allows traffic only through specific ports, locking down communication with the instance according to the rules that you specify. Using an Elastic IP address (EIP) enables an instance in a VPC, which is otherwise private, to be reached from the Internet through an Internet gateway (for example, it could act as a web server).



HINT We don't create Elastic IP(EIP) in this exercise. Our VPC and Subnets are also slightly different CIDR Blocks.

What is Amazon VPC?

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Introduction.html

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Topics
Amazon VPC Concepts
How to Get Started with Amazon VPC
Using Amazon VPC with Other AWS Services

Accessing Amazon VPC
Pricing for Amazon VPC
Amazon VPC Limits
PCI DSS Compliance

Amazon VPC Concepts

As you get started with Amazon VPC, you should understand the key concepts of this virtual network, and how it is similar to or different from your own networks. This section provides a brief description of the key concepts for Amazon VPC.

Amazon VPC is the networking layer for Amazon EC2. If you're new to Amazon EC2, see What is Amazon EC2? in the Amazon EC2 User Guide for Linux Instances to get a brief overview.

VPCs and Subnets

A **virtual private cloud** (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings.

A **subnet** is a range of IP addresses in your VPC. You can launch AWS resources into a subnet that you select. Use a public subnet for resources that must be connected to the Internet, and a private subnet for resources that won't be connected to the Internet. For more information about public and private subnets, see VPC and Subnet Basics.

To protect the AWS resources in each subnet, you can use multiple layers of security, including security groups and network access control lists (ACL). For more information, see Security.

Supported Platforms

The original release of Amazon EC2 supported a single, flat network that's shared with other customers called the EC2-Classic platform. Older AWS accounts still support this platform, and can launch instances into either EC2-Classic or a VPC. Accounts created after 2013-12-04 support EC2-VPC only. For more information, see Detecting Your Supported Platforms and Whether You Have a Default VPC.

By launching your instances into a VPC instead of EC2-Classic, you gain the ability to:

- Assign static private IPv4 addresses to your instances that persist across starts and stops
- Optionally associate an IPv6 CIDR block to your VPC and assign IPv6 addresses to your instances
- Assign multiple IP addresses to your instances
- Define network interfaces, and attach one or more network interfaces to your instances
- Change security group membership for your instances while they're running
- Control the outbound traffic from your instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)
- Add an additional layer of access control to your instances in the form of network access control lists (ACL)
- Run your instances on single-tenant hardware

Default and Nondefault VPCs

If your account supports the EC2-VPC platform only, it comes with a default VPC that has a default subnet in each Availability Zone. A default VPC has the benefits of the advanced features provided by EC2-VPC, and is ready for you to use. If you have a default VPC and don't specify a subnet when you launch an instance, the

instance is launched into your default VPC. You can launch instances into your default VPC without needing to know anything about Amazon VPC.

Regardless of which platforms your account supports, you can create your own VPC, and configure it as you need. This is known as a nondefault VPC. Subnets that you create in your nondefault VPC and additional subnets that you create in your default VPC are called nondefault subnets.

Accessing the Internet
You control how the instances that you launch into a VPC access resources outside the VPC.

Your default VPC includes an Internet gateway, and each default subnet is a public subnet. Each instance that you launch into a default subnet has a private IPv4 address and a public IPv4 address. These instances can communicate with the Internet through the Internet gateway. An Internet gateway enables your instances to connect to the Internet through the Amazon EC2 network edge.

By default, each instance that you launch into a nondefault subnet has a private IPv4 address, but no public IPv4 address, unless you specifically assign one at launch, or you modify the subnet's public IP address attribute. These instances can communicate with each other, but can't access the Internet.

You can enable Internet access for an instance launched into a nondefault subnet by attaching an Internet gateway to its VPC (if its VPC is not a default VPC) and associating an Elastic IP address with the instance.

How to Get Started with Amazon VPC

To get a hands-on introduction to Amazon VPC, complete the exercise Getting Started. The exercise will guide you through the steps to create a nondefault VPC with a public subnet, and to launch an instance into your subnet.

If you have a default VPC, and you want to get started launching instances into your VPC without performing any additional configuration on your VPC, see Launching an EC2 Instance into Your Default VPC.

To learn about the basic scenarios for Amazon VPC, see Scenarios and Examples. You can configure your VPC and subnets in other ways to suit your needs.

The following table lists related resources that you'll find useful as you work with this service.

Resource	Description
Amazon Virtual Private Cloud Connectivity Options	A whitepaper that provides an overview of the options for network connectivity.
Amazon VPC forum	A community-based forum for discussing technical questions related to Amazon VPC.
AWS Developer Resources	A central starting point to find documentation, code samples, release notes, and other information to help you create innovative applications with AWS.
AWS Support Center	The home page for AWS Support.
Contact Us	A central contact point for inquiries concerning AWS billing, accounts, and events.

Using Amazon VPC with Other AWS Services

Amazon VPC integrates with many other AWS services; furthermore, some services require a VPC in your account to carry out certain functions. Below are examples of services that use Amazon VPC.

Service	Relevant Topic
AWS Data Pipeline	Launching Resources for Your Pipeline into a VPC

Amazon EC2	Amazon EC2 and Amazon VPC
Auto Scaling	Auto Scaling and Amazon VPC
Elastic Beanstalk	Using AWS Elastic Beanstalk with Amazon VPC
Elastic Load Balancing	Setting Up Elastic Load Balancing
Amazon ElastiCache	Using ElastiCache with Amazon VPC
Amazon EMR	Select a Subnet for the Cluster
AWS OpsWorks	Running a Stack in a VPC
Amazon RDS	Amazon RDS and Amazon VPC
Amazon Redshift	Managing Clusters in a VPC
Amazon Route 53	Working with Private Hosted Zones
Amazon WorkSpaces	Create and Configure Your VPC

To get a detailed view of the VPCs, subnets, and other VPC resources in your account and their relation to each other, you can use the AWS Config service. For more information, see What is AWS Config? in the AWS Config Developer Guide.

Accessing Amazon VPC

Amazon VPC provides a web-based user interface, the Amazon VPC console. If you've signed up for an AWS account, you can access the Amazon VPC console by signing into the AWS Management Console and selecting **VPC** from the console home page.

If you prefer to use a command line interface, you have the following options:

AWS Command Line Interface (CLI)

Provides commands for a broad set of AWS products, and is supported on Windows, Mac, and Linux/UNIX. To get started, see AWS Command Line Interface User Guide. For more information about the commands for Amazon VPC, see ec2.

AWS Tools for Windows PowerShell

Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see AWS Tools for Windows PowerShell User Guide.

Pricing for Amazon VPC

There's no additional charge for using Amazon VPC. You pay the standard rates for the instances and other Amazon EC2 features that you use.

Amazon VPC Limits

There are limits to the number of Amazon VPC components that you can provision. You can request an increase for some of these limits. For more information, see Amazon VPC Limits.

PCI DSS Compliance

Amazon VPC supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS). For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see PCI DSS Level 1.

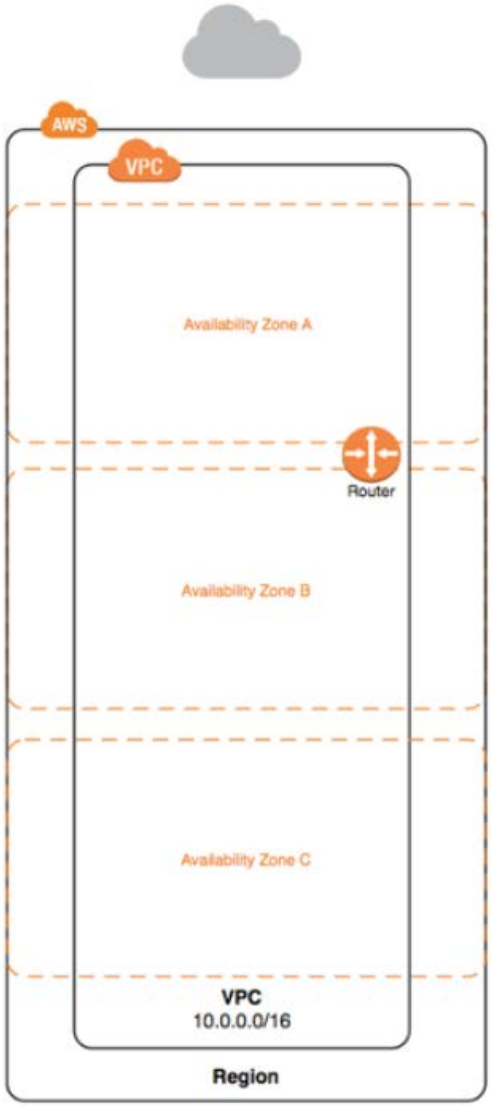
VPC and Subnet Basics

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#vpc-subnet-basics

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.

When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, `10.0.0.0/16`. For more information about CIDR notation, see RFC 4632.

The following diagram shows a new VPC with an IPv4 CIDR block, and the main route table.



Main route table

Destination	Target
10.0.0.0/16	local

When you create a VPC, it spans all the Availability Zones in the region. After creating a VPC, you can add one or more subnets in each Availability Zone. When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block. Each subnet must reside entirely within one Availability Zone and cannot span zones. Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location. We assign a unique ID to each subnet.

If a subnet's traffic is routed to an Internet gateway, the subnet is known as a **public subnet**. In this diagram, subnet 1 is a public subnet. If you want your instance in a public subnet to communicate with the Internet over IPv4, it must have a public IPv4 address or an Elastic IP address (IPv4). For more information about public IPv4 addresses, see Public IPv4 Addresses. If you want your instance in the public subnet to communicate with the Internet over IPv6, it must have an IPv6 address.

If a subnet doesn't have a route to the Internet gateway, the subnet is known as a **private subnet**. In this diagram, subnet 2 is a private subnet.

For more information, see Scenarios and Examples, Internet Gateways, or Adding a Hardware Virtual Private Gateway to Your VPC.

> **Note**
> Regardless of the type of subnet, the internal IPv4 address range of the subnet is always private—we do not announce the address block to the Internet.

You have a limit on the number of VPCs and subnets you can create in your account. For more information, see Amazon VPC Limits.

VPC and Subnet Sizing

Amazon VPC supports IPv4 and IPv6 addressing, and has different CIDR block size limits for each. By default, all VPCs and subnets must have IPv4 CIDR blocks—you can't change this behavior. You can choose whether to associate an IPv6 CIDR block with your VPC.

VPC and Subnet Sizing for IPv4

You can assign a single CIDR block to a VPC. The allowed block size is between a /16 netmask and /28 netmask. In other words, the VPC can contain from 16 to 65,536 IP addresses.

When you create a VPC, we recommend that you specify a CIDR block from the private (non-publicly routable) IPv4 address ranges as specified in RFC 1918:

`10.0.0.0` - `10.255.255.255` (10/8 prefix)
`172.16.0.0` - `172.31.255.255` (172.16/12 prefix)
`192.168.0.0` - `192.168.255.255` (192.168/16 prefix)

You can create a VPC with a publicly routable CIDR block that falls outside of the private IPv4 address ranges specified in RFC 1918; however, for the purposes of this documentation, we refer to private IP addresses as the IPv4 addresses that are within the CIDR range of your VPC.

You can't change the size of a VPC after you create it. If your VPC is too small to meet your needs, create a new, larger VPC, and then migrate your instances to the new VPC. To do this, create AMIs from your running instances, and then launch replacement instances in your new, larger VPC. You can then terminate your old instances, and delete your smaller VPC. For more information, see Deleting Your VPC.

The CIDR block of a subnet can be the same as the CIDR block for the VPC (for a single subnet in the VPC), or a subset (for multiple subnets). The allowed block size is between a `/28` netmask and `/16` netmask. If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap.

For example, if you create a VPC with CIDR block `10.0.0.0/24`, it supports 256 IP addresses. You can break this CIDR block into two subnets, each supporting 128 IP addresses. One subnet uses CIDR block `10.0.0.0/25` (for addresses `10.0.0.0` - `10.0.0.127`) and the other uses CIDR block `10.0.0.128/25` (for addresses `10.0.0.128` - `10.0.0.255`).

There are many tools available to help you calculate subnet CIDR blocks; for example, see http://www.subnet-calculator.com/cidr.php. Also, your network engineering group can help you determine the CIDR blocks to specify for your subnets.

The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance. For example, in a subnet with CIDR block `10.0.0.0/24`, the following five IP addresses are reserved:

- `10.0.0.0`: Network address.
- `10.0.0.1`: Reserved by AWS for the VPC router.
- `10.0.0.2`: Reserved by AWS. The IP address of the DNS server is always the base of the VPC network range plus two; however, we also reserve the base of each subnet range plus two. For more information, see Amazon DNS Server.
- `10.0.0.3`: Reserved by AWS for future use.
- `10.0.0.255`: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

Subnet Routing

Each subnet must be associated with a route table, which specifies the allowed routes for outbound traffic leaving the subnet. Every subnet that you create is automatically associated with the main route table for the VPC. You can change the association, and you can change the contents of the main route table. For more information, see Route Tables.

In the previous diagram, the route table associated with subnet 1 routes all IPv4 traffic (`0.0.0.0/0`) and IPv6 traffic (`::/0`) to an Internet gateway (for example, `igw-1a2b3c4d`). Because instance 1A has an IPv4 Elastic IP address and instance 1B has an IPv6 address, they can be reached from the Internet over IPv4 and IPv6 respectively.

Note

(IPv4 only) The Elastic IPv4 address or public IPv4 address that's associated with your instance is accessed through the Internet gateway of your VPC.

Subnet Security

AWS provides two features that you can use to increase security in your VPC: security groups and network ACLs. Security groups control inbound and outbound traffic for your instances, and network ACLs control inbound and outbound traffic for your subnets. In most cases, security groups can meet your needs; however, you can also use network ACLs if you want an additional layer of security for your VPC. For more information, see Security.

By design, each subnet must be associated with a network ACL. Every subnet that you create is automatically associated with the VPC's default network ACL. You can change the association, and you can change the contents of the default network ACL. For more information, see Network ACLs.

You can create a flow log on your VPC or subnet to capture the traffic that flows to and from the network interfaces in your VPC or subnet. You can also create a flow log on an individual network interface. Flow logs are published to CloudWatch Logs. For more information, see VPC Flow Logs.

Working with VPCs and Subnets

You can create a VPC and subnets using the Amazon VPC console. The following procedures are for manually creating a VPC and subnets. You also have to manually add gateways and routing tables. Alternatively, you can use the Amazon VPC wizard to create a VPC plus its subnets, gateways, and routing tables in one step. For more information, see Scenarios and Examples.

Internet Gateways

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html

An Internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

An Internet gateway serves two purposes: to provide a target in your VPC route tables for Internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

An Internet gateway supports IPv4 and IPv6 traffic.

Enabling Internet Access

To enable access to or from the Internet for instances in a VPC subnet, you must do the following:

- Attach an Internet gateway to your VPC.
- Ensure that your subnet's route table points to the Internet gateway.
- Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).
- Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance.

To use an Internet gateway, your subnet's route table must contain a route that directs Internet-bound traffic to the Internet gateway. You can scope the route to all destinations not explicitly known to the route table (0.0.0.0/0 for IPv4 or ::/0 for IPv6), or you can scope the route to a narrower range of IP addresses; for example, the public IPv4 addresses of your company's public endpoints outside of AWS, or the Elastic IP addresses of other Amazon EC2 instances outside your VPC. If your subnet is associated with a route table that has a route to an Internet gateway, it's known as a public subnet.

To enable communication over the Internet for IPv4, your instance must have a public IPv4 address or an Elastic IP address that's associated with a private IPv4 address on your instance. Your instance is only aware of the private (internal) IP address space defined within the VPC and subnet. The Internet gateway logically provides the one-to-one NAT on behalf of your instance, so that when traffic leaves your VPC subnet and goes to the Internet, the reply address field is set to the public IPv4 address or Elastic IP address of your instance, and not its private IP address. Conversely, traffic that's destined for the public IPv4 address or Elastic IP address of your instance has its destination address translated into the instance's private IPv4 address before the traffic is delivered to the VPC.

Internet Access for Default and Nondefault VPCs
The following table provides an overview of whether your VPC automatically comes with the components required for Internet access over IPv4 or IPv6.

	Default VPC	Nondefault VPC
Internet gateway	Yes	Yes, if you created the VPC using the first or second option in the VPC wizard. Otherwise, you must manually create and attach the Internet gateway.
Route table with route to Internet gateway for IPv4 traffic (0.0.0.0/0)	Yes	Yes, if you created the VPC using the first or second option in the VPC wizard. Otherwise, you must manually create the route table and add the route.
Route table with route to Internet gateway for IPv6 traffic (::/0)	No	Yes, if you created the VPC using the first or second option in the VPC wizard, and if you specified the option to associate an IPv6 CIDR block with the VPC. Otherwise, you must manually create the route table and add the route.
Public IPv4 address automatically assigned to instance launched into subnet	Yes (default subnet)	No (nondefault subnet)
IPv6 address automatically assigned to instance launched into subnet	No (default subnet)	No (nondefault subnet)

For more information about default VPCs, see Default VPC and Default Subnets. For more information about using the VPC wizard to create a VPC with an Internet gateway, see Scenario 1: VPC with a Single Public Subnet or Scenario 2: VPC with Public and Private Subnets (NAT).

For more information about IP addressing in your VPC, and controlling how instances are assigned public IPv4 or IPv6 addresses, see IP Addressing in Your VPC.

When you add a new subnet to your VPC, you must set up the routing and security that you want for the subnet.

Route Tables

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Route_Tables.html

A route table contains a set of rules, called routes, that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

Topics
Route Table Basics
Route Priority
Routing Options

Route Table Basics

The following are the basic things that you need to know about route tables:

- Your VPC has an implicit router.
- Your VPC automatically comes with a main route table that you can modify.
- You can create additional custom route tables for your VPC.
- Each subnet must be associated with a route table, which controls the routing for the subnet. If you don't explicitly associate a subnet with a particular route table, the subnet is implicitly associated with the main route table.

- You cannot delete the main route table, but you can replace the main route table with a custom table that you've created (so that this table is the default table each new subnet is associated with).
- Each route in a table specifies a destination CIDR and a target (for example, traffic destined for the external corporate network 172.16.0.0/12 is targeted for the virtual private gateway). We use the most specific route that matches the traffic to determine how to route the traffic.
- CIDR blocks for IPv4 and IPv6 are treated separately. For example, a route with a destination CIDR of 0.0.0.0/0 (all IPv4 addresses) does not automatically include all IPv6 addresses. You must create a route with a destination CIDR of ::/0 for all IPv6 addresses.
- Every route table contains a local route for communication within the VPC over IPv4. If you've associated an IPv6 CIDR block with your VPC, every route table also contains a local route for communication within the VPC over IPv6. You cannot modify or delete these routes.
- When you add an Internet gateway, an egress-only Internet gateway, a virtual private gateway, a NAT device, a peering connection, or a VPC endpoint in your VPC, you must update the route table for any subnet that uses these gateways or connections.
- There is a limit on the number of route tables you can create per VPC, and the number of routes you can add per route table. For more information, see Amazon VPC Limits.

Main Route Tables
When you create a VPC, it automatically has a main route table. On the Route Tables page in the Amazon VPC console, you can view the main route table for a VPC by looking for **Yes** in the Main column. The main route table controls the routing for all subnets that are not explicitly associated with any other route table. You can add, remove, and modify routes in the main route table.

You can explicitly associate a subnet with the main route table, even if it's already implicitly associated. You might do that if you change which table is the main route table, which changes the default for additional new subnets, or any subnets that are not explicitly associated with any other route table. For more information, see Replacing the Main Route Table.

Custom Route Tables
Your VPC can have route tables other than the default table. One way to protect your VPC is to leave the main route table in its original default state (with only the local route), and explicitly associate each new subnet you create with one of the custom route tables you've created. This ensures that you explicitly control how each subnet routes outbound traffic.

If you create a new subnet in this VPC, it's automatically associated with the main route table.

Route Table Association
The VPC console shows the number of subnets explicitly associated with each route table, and provides information about subnets that are implicitly associated with the main route table. For more information, see Determining Which Subnets Are Explicitly Associated with a Table).

Subnets can be implicitly or explicitly associated with the main route table. Subnets typically won't have an explicit association to the main route table, although it might happen temporarily if you're replacing the main route table.

You might want to make changes to the main route table, but to avoid any disruption to your traffic, you can first test the route changes using a custom route table. After you're satisfied with the testing, you then replace the main route table with the new custom table.

The following diagram shows a VPC with two subnets that are implicitly associated with the main route table (Route Table A), and a custom route table (Route Table B) that isn't associated with any subnets.

You can create an explicit association between Subnet 2 and Route Table B.

After you've tested Route Table B, you can make it the main route table. Note that Subnet 2 still has an explicit association with Route Table B, and Subnet 1 has an implicit association with Route Table B because it is the new main route table. Route Table A is no longer in use.

If you disassociate Subnet 2 from Route Table B, there's still an implicit association between Subnet 2 and Route Table B. If you no longer need Route Table A, you can delete it.

Route Priority

We use the most specific route in your route table that matches the traffic to determine how to route the traffic (longest prefix match).

Routes to IPv4 and IPv6 addresses or CIDR blocks are independent of each other; we use the most specific route that matches either IPv4 traffic or IPv6 traffic to determine how to route the traffic.

For example, the following route table has a route for IPv4 Internet traffic (`0.0.0.0/0`) that points to an Internet gateway, and a route for `172.31.0.0/16` IPv4 traffic that points to a peering connection (`pcx-1a2b3c4d`). Any traffic from the subnet that's destined for the `172.31.0.0/16` IP address range uses the peering connection, because this route is more specific than the route for Internet gateway. Any traffic destined for a target within the VPC (`10.0.0.0/16`) is covered by the `Local` route, and therefore routed within the VPC. All other traffic from the subnet uses the Internet gateway.

Destination	Target
10.0.0.0/16	Local
172.31.0.0/16	pcx-1a2b1a2b
0.0.0.0/0	igw-11aa22bb

Routing Options

The following topics explain routing for specific gateways or connections in your VPC.

Topics

Route Tables for an Internet Gateway

Route Tables for an Internet Gateway

You can make a subnet a public subnet by adding a route to an Internet gateway. To do this, create and attach an Internet gateway to your VPC, and then add a route with a destination of `0.0.0.0/0` for IPv4 traffic or `::/0` for IPv6 traffic, and a target of the Internet gateway ID (`igw-xxxxxxxx`). For more information, see Internet Gateways.

Security Groups for Your VPC

URL Link http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign the instance to up to five security groups. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups. If you don't specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

For each security group, you add rules that control the inbound traffic to instances, and a separate set of rules that control the outbound traffic. This section describes the basics things you need to know about security groups for your VPC and their rules.

You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC. For more information about the differences between security groups and network ACLs, see Comparison of Security Groups and Network ACLs.

Topics

Security Group Basics

The following are the basic characteristics of security groups for your VPC:

- You have limits on the number of security groups that you can create per VPC, the number of rules that you can add to each security group, and the number of security groups you can associate with a network interface. For more information, see Amazon VPC Limits.
- You can specify allow rules, but not deny rules.
- You can specify separate rules for inbound and outbound traffic.
- When you create a security group, it has no inbound rules. Therefore, no inbound traffic is allowed until you add inbound rules to the security group.
- By default, a security group includes an outbound rule that allows all outbound traffic. You can remove the rule and add outbound rules that allow specific outbound traffic only. If your security group has no outbound rules, no outbound traffic is allowed.
- Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.
    - Note Some types of traffic are tracked differently to others. For more information, see Connection Tracking in the Amazon EC2 User Guide for Linux Instances.
- Instances associated with a security group can't talk to each other unless you add rules allowing it (exception: the default security group has these rules by default).
- Security groups are associated with network interfaces. After you launch an instance, you can change the security groups associated with the instance, which changes the security groups associated with the primary network interface (eth0). You can also change the security groups associated with any other network interface. For more information about network interfaces, see Elastic Network Interfaces.

Default Security Group for Your VPC

Your VPC automatically comes with a default security group. Each EC2 instance that you launch in your VPC is automatically associated with the default security group if you don't specify a different security group when you launch the instance.

The following table describes the default rules for a default security group.

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The security group ID (sg-xxxxxxxx)	All	All	Allow inbound traffic from instances assigned to the same security group.
Outbound			
Destination	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	All	All	Allow all outbound IPv4 traffic.
::/0	All	All	Allow all outbound IPv6 traffic. This rule is added by default if you create a VPC with an IPv6 CIDR block or if you associate an IPv6 CIDR block with your existing VPC.

You can change the rules for the default security group.
You can't delete a default security group. If you try to delete the default security group, you'll get the following error: `Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user.`

Security Group Rules

You can add or remove rules for a security group (also referred to as authorizing or revoking inbound or outbound access). A rule applies either to inbound traffic (ingress) or outbound traffic (egress). You can grant access to a specific CIDR range, or to another security group in your VPC or in a peer VPC (requires a VPC peering connection).

The following are the basic parts of a security group rule in a VPC:

- (Inbound rules only) The source of the traffic and the destination port or port range. The source can be another security group, an IPv4 or IPv6 CIDR block, or a single IPv4 or IPv6 address.
- (Outbound rules only) The destination for the traffic and the destination port or port range. The destination can be another security group, an IPv4 or IPv6 CIDR block, or a single IPv4 or IPv6 address.
- Any protocol that has a standard protocol number (for a list, see Protocol Numbers). If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.

When you specify a security group as the source for a rule, this allows instances associated with the source security group to access instances in the security group. (Note that this does not add rules from the source security group to this security group.)

If you specify a single IPv4 address, use the /32 prefix. If you specify a single IPv6 address, specify the /128 prefix length.

Some systems for setting up firewalls let you filter on source ports. Security groups let you filter only on destination ports.

When you add or remove rules, they are automatically applied to all instances associated with the security group.

The kind of rules you add may depend on the purpose of the instance. The following table describes example rules for a security group for web servers. The web servers can receive HTTP and HTTPS traffic from all IPv4 and IPv6 addresses, and send SQL or MySQL traffic to a database server.

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	TCP	80	Allow inbound HTTP access from all IPv4 addresses
::/0	TCP	80	Allow inbound HTTP access from all IPv6 addresses
0.0.0.0/0	TCP	443	Allow inbound HTTPS access from all IPv4 addresses
::/0	TCP	443	Allow inbound HTTPS access from all IPv6 addresses
Your network's public IPv4 address range	TCP	22	Allow inbound SSH access to Linux instances from IPv4 IP addresses in your network (over the Internet gateway)
Your network's public IPv4 address range	TCP	3389	Allow inbound RDP access to Windows instances from IPv4 IP addresses in your network (over the Internet gateway)
Outbound			
Destination	**Protocol**	**Port Range**	**Comments**
The ID of the security group for your database servers	TCP	1433	Allow outbound Microsoft SQL Server access to instances in the specified security group
The ID of the security group for your MySQL database servers	TCP	3306	Allow outbound MySQL access to instances in the specified security group

A database server would need a different set of rules; for example, instead of inbound HTTP and HTTPS traffic, you can add a rule that allows inbound MySQL or Microsoft SQL Server access. For an example of security group rules for web servers and database servers, see Security.

For more information about creating security group rules to ensure that Path MTU Discovery can function correctly, see Path MTU Discovery in the Amazon EC2 User Guide for Linux Instances.

Differences Between Security Groups for EC2-Classic and EC2-VPC

If you're already an Amazon EC2 user, you're probably familiar with security groups. However, you can't use the security groups that you've created for use with EC2-Classic with instances in your VPC. You must create security groups specifically for use with instances in your VPC. The rules you create for use with a security group for a VPC can't reference a security group for EC2-Classic, and vice versa.

The following table summarizes the differences between security groups for use with EC2-Classic and those for use with EC2-VPC.

EC2-Classic	EC2-VPC
You can create up to 500 security groups per region.	You can create up to 500 security groups per VPC.
You can add up to 100 rules to a security group.	You can add up to 50 rules to a security group.
You can add rules for inbound traffic only.	You can add rules for inbound and outbound traffic.
You can assign up to 500 security groups to an instance.	You can assign up to 5 security groups to a network interface.
You can reference security groups from other AWS accounts.	You can reference security groups from your VPC or from a peer VPC in a VPC peering connection only. The peer VPC can be in a different account.
After you launch an instance, you can't change the security groups assigned to it.	You can change the security groups assigned to an instance after it's launched.
When you add a rule to a security group, you don't have to specify a protocol, and only TCP, UDP, or ICMP are available.	When you add a rule to a security group, you must specify a protocol, and it can be any protocol with a standard protocol number, or all protocols (see Protocol Numbers).
When you add a rule to a security group, you must specify port numbers (for TCP or UDP).	When you add a rule to a security group, you can specify port numbers only if the rule is for TCP or UDP, and you can specify all port numbers.
Security groups that are referenced in another security group's rules cannot be deleted.	Security groups that are referenced in another security group's rules can be deleted if the security groups are in different VPCs. If the referenced security group is deleted, the rule is marked as stale. You can use the describe-stale-security-groups AWS CLI command to identify stale rules.
You cannot specify an IPv6 CIDR block or an IPv6 address as the source or destination in a security group rule.	You can specify an IPv6 CIDR block or an IPv6 address as the source or destination in a security group rule.

Working with Security Groups

This section shows you how to work with security groups using the Amazon VPC console.

Topics

Modifying the Default Security Group

Your VPC includes a default security group whose initial rules are to deny all inbound traffic, allow all outbound traffic, and allow all traffic between instances in the group. You can't delete this group; however, you can change the group's rules. The procedure is the same as modifying any other security group. For more information, see Adding and Removing Rules.

Creating a Security Group

Although you can use the default security group for your instances, you might want to create your own groups to reflect the different roles that instances play in your system.

Adding and Removing Rules

When you add or remove a rule, any instances already assigned to the security group are subject to the change.

Changing an Instance's Security Groups

You can change the security groups that an instance in a VPC is assigned to after the instance is launched. When you make this change, the instance can be either running or stopped.

> **Note**
> This procedure changes the security groups that are associated with the primary network interface (eth0) of the instance. To change the security groups for other network interfaces, see Changing the Security Group of a Network Interface.

Deleting a Security Group

You can delete a security group only if there are no instances assigned to it (either running or stopped). You can assign the instances to another security group before you delete the security group (see Changing an Instance's Security Groups). You can't delete a default security group.

Deleting the 2009-07-15-default Security Group

Any VPC created using an API version older than 2011-01-01 has the 2009-07-15-default security group. This security group exists in addition to the regular default security group that comes with every VPC. You can't attach an Internet gateway to a VPC that has the 2009-07-15-default security group. Therefore, you must delete this security group before you can attach an Internet gateway to the VPC.

> **Note**
> If you assigned this security group to any instances, you must assign these instances a different security group before you can delete the security group.

Identity and Access Management (IAM)

Creating, Modifying and Viewing Access Keys (AWS Management Console)

URL Link http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

Users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services. To fill this need, you can create, modify, view, or rotate access keys (access key IDs and secret access keys) for IAM users.

When you create an access key, IAM returns the access key ID and secret access key. You should save these in a secure location and give them to the user.

> **Important**
> To ensure the security of your AWS account, the secret access key is accessible only at the time you create it. If a secret access key is lost, you must delete the access key for the associated user and create a new key. For more details, see Retrieving Your Lost or Forgotten Passwords or Access Keys.

By default, when you create an access key, its status is Active, which means the user can use the access key for AWS CLI, Tools for Windows PowerShell, and API calls. Each user can have two active access keys, which is useful when you must rotate the user's access keys. You can disable a user's access key, which means it can't be used for API calls. You might do this while you're rotating keys or to revoke API access for a user.
You can delete an access key at any time. However, when you delete an access key, it's gone forever and cannot be retrieved. (You can always create new keys.)

IAM Roles

URL Link http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html

An IAM role is similar to a user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is

intended to be assumable by anyone who needs it. Also, a role does not have any credentials (password or access keys) associated with it. Instead, if a user is assigned to a role, access keys are created dynamically and provided to the user.

You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources. For example, you might want to grant users in your AWS account access to resources they don't usually have, or grant users in one AWS account access to resources in another account. Or you might want to allow a mobile app to use AWS resources, but not want to embed AWS keys within the app (where they can be difficult to rotate and where users can potentially extract them). Sometimes you want to give AWS access to users who already have identities defined outside of AWS, such as in your corporate directory. Or, you might want to grant access to your account to third parties so that they can perform an audit on your resources.

For these scenarios, you can delegate access to AWS resources using an IAM role. This section introduces roles and the different ways you can use them, when and how to choose among approaches, and how to create, manage, switch to (or assume), and delete roles.

Elastic Compute Cloud (EC2)

Instance Metadata

URL Link http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories. For more information, see Instance Metadata Categories.
EC2 instances can also include dynamic data, such as an instance identity document that is generated when the instance is launched. For more information, see Dynamic Data Categories.

You can also access the user data that you supplied when launching your instance. For example, you can specify parameters for configuring your instance, or attach a simple script. You can also use this data to build more generic AMIs that can be modified by configuration files supplied at launch time. For example, if you run web servers for various small businesses, they can all use the same AMI and retrieve their content from the Amazon S3 bucket you specify in the user data at launch. To add a new customer at any time, simply create a bucket for the customer, add their content, and launch your AMI. If you launch more than one instance at the same time, the user data is available to all instances in that reservation.

Important

Although you can only access instance metadata and user data from within the instance itself, the data is not protected by cryptographic methods. Anyone who can access the instance can view its metadata. Therefore, you should take suitable precautions to protect sensitive data (such as long-lived encryption keys). You should not store sensitive data, such as passwords, as user data.

Contents

Because your instance metadata is available from your running instance, you do not need to use the Amazon EC2 console or the AWS CLI. This can be helpful when you're writing scripts to run from your instance. For example, you can access the local IP address of your instance from instance metadata to manage a connection to an external application.

To view all categories of instance metadata from within a running instance, use the following URI:

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

Note that you are not billed for HTTP requests used to retrieve instance metadata and user data.

You can use a tool such as cURL, or if your instance supports it, the GET command; for example:

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

This example gets the top-level metadata items. Some items are only available for instances in a VPC. For more information about each of these items, see Instance Metadata Categories.

```
$ curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
kernel-id
local-hostname
local-ipv4
mac
network/
placement/
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

These examples get the value of some of the metadata items from the preceding example.

```
$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-12345678
$ curl http://169.254.169.254/latest/meta-data/reservation-id
r-fea54097
$ curl http://169.254.169.254/latest/meta-data/local-hostname
ip-10-251-50-12.ec2.internal
$ curl http://169.254.169.254/latest/meta-data/public-hostname
ec2-203-0-113-25.compute-1.amazonaws.com
```

This example gets the list of available public keys.

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/
0=my-public-key
```

This example shows the formats in which public key 0 is available.

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/0/
openssh-key
```

This example gets public key 0 (in the OpenSSH key format).

```
$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-
key
ssh-rsa
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
```

```
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

This example gets the subnet ID for an instance launched into a VPC.

```
$ curl http://169.254.169.254/latest/meta-
data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

AWS Command Line Interface (CLI)

What Is the AWS Command Line Interface?

URL Link http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html

The AWS Command Line Interface is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

HINT Using CLI requires some scripting / development skill sets. We won't go very deep in any initial lessons. However, it is important that you understand the CLI exists, how to access it, and have some familiarity with the AWS CLI commands for the core services (S3, EC2, RDS, VPC, etc.). More on this later!

Configuring the AWS Command Line Interface

URL Link http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html

This section explains how to configure settings that the AWS Command Line Interface uses when interacting with AWS, such as your security credentials and the default region.

Sections

Quick Configuration
Configuration and Credential Files
Instance Metadata

Quick Configuration

For general use, the `aws configure` command is the fastest way to set up your AWS CLI installation.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

The AWS CLI will prompt you for four pieces of information. AWS Access Key ID and AWS Secret Access Key are your account credentials. If you don't have keys, see the Getting Set Up section earlier in this guide.

Default region is the name of the region you want to make calls against by default. This is usually the region closest to you, but it can be any region.

> **Note**
> You must specify an AWS region when using the AWS CLI. For a list of services and available regions, see Regions and Endpoints.

Default output format can be either json, text, or table. If you don't specify an output format, json will be used.

To update any of your settings, simply run `aws configure` again and enter new values as appropriate. The next sections contains more information on the files that `aws configure` creates, additional settings, and named profiles.

Configuration and Credential Files

The CLI stores credentials specified with `aws configure` in a local file named `credentials` in a folder named `.aws` in your home directory. Home directory location varies but can be referred to using the environment variables `%UserProfile%` in Windows and `$HOME` or `~` (tilde) in Unix-like systems.
For example, the following commands list the contents of the `.aws` folder:

Linux, macOS, or Unix

```
$ ls  ~/.aws
```

Windows

```
> dir %UserProfile%\.aws
```

In order to separate credentials from less sensitive options, region and output format are stored in a separate file named config in the same folder.

The default file location for the config file can be overridden by setting the AWS_CONFIG_FILE environment variable to another local path. See Environment Variables for details.

> **Storing Credentials in Config**
> The AWS CLI will also read credentials from the config file. If you want to keep all of your profile settings in a single file, you can. If there are ever credentials in both locations for a profile (say you used aws configure to update the profile's keys), the keys in the credentials file will take precedence.
> If you use one of the SDKs in addition to the AWS CLI, you may notice additional warnings if credentials are not stored in their own file.

The files generated by the CLI for the profile configured in the previous section look like this:

`~/.aws/credentials`

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

`~/.aws/config`

```
[default]
region=us-west-2
output=json
```

The following settings are supported.

aws_access_key_id – AWS access key.
aws_secret_access_key – AWS secret key.
aws_session_token – AWS session token. A session token is only required if you are using temporary security credentials.
region – AWS region.
output – output format (json, text, or table)

Instance Metadata

To use the CLI from an EC2 instance, create a role that has access to the resources needed and assign that role to the instance when it is launched. Launch the instance and check to see if the AWS CLI is already installed (it comes pre-installed on Amazon Linux).

Install the AWS CLI if necessary and configure a default region to avoid having to specify it in every command. You can set the region using aws configure without entering credentials by pressing enter twice to skip the first two prompts:

```
$ aws configure
AWS Access Key ID [None]: ENTER
AWS Secret Access Key [None]: ENTER
Default region name [None]: us-west-2
Default output format [None]: json
```

The AWS CLI will read credentials from the instance metadata. For more information, see Granting Applications that Run on Amazon EC2 Instances Access to AWS Resources in IAM User Guide.

Relational Database Services (RDS)

Working with an Amazon RDS DB Instance in a VPC

URL Link http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.WorkingWithRDSInstanceinaVPC.html

Unless you are working with a legacy DB instance, your DB instance is in a virtual private cloud (VPC). A virtual private cloud is a virtual network that is logically isolated from other virtual networks in the AWS cloud. Amazon Virtual Private Cloud (Amazon VPC) lets you launch AWS resources, such as an Amazon Relational Database Service (Amazon RDS) or Amazon Elastic Compute Cloud (Amazon EC2) instance, into a VPC. The VPC can either be a default VPC that comes with your account or one that you create. All VPCs are associated with your AWS account.

Your default VPC has three subnets you can use to isolate resources inside the VPC. The default VPC also has an Internet Gateway that can be used to provide access to resources inside the VPC from outside the VPC.
For a list of scenarios involving Amazon RDS DB instances in a VPC and outside of a VPC, see Scenarios for Accessing a DB Instance in a VPC.

For a tutorial that shows you how to create a VPC that you can use with a common Amazon RDS scenario, see Tutorial: Create an Amazon VPC for Use with an Amazon RDS DB Instance.

To learn how to work with an Amazon RDS DB instances inside a VPC, see the following:
Topics
Working with a DB Instance in a VPC
Working with DB Subnet Groups
Hiding a DB Instance in a VPC from the Internet
Creating a DB Instance in a VPC
Updating the VPC for a DB Instance

Working with a DB Instance in a VPC

Here are some tips on working with a DB instance in a VPC:

- Your VPC must have at least one subnet in at least two of the Availability Zones in the region where you want to deploy your DB instance. A subnet is a segment of a VPC's IP address range that you can specify and that lets you group instances based on your security and operational needs.
- If you want your DB instance in the VPC to be publicly accessible, you must enable the VPC attributes DNS hostnames and DNS resolution.
- Your VPC must have a DB subnet group that you create (for more information, see the next section). You create a DB subnet group by specifying the subnets you created. Amazon RDS uses that DB subnet group and your preferred Availability Zone to select a subnet and an IP address within that subnet to assign to your DB instance.
- Your VPC must have a VPC security group that allows access to the DB instance.
- The CIDR blocks in each of your subnets must be large enough to accommodate spare IP addresses for Amazon RDS to use during maintenance activities, including failover and compute scaling.
-

Working with DB Subnet Groups

Subnets are segments of a VPC's IP address range that you designate to group your resources based on security and operational needs. A DB subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when creating DB instances using the CLI or API; if you use the console, you can just select the VPC and subnets you want to use.

Each DB subnet group should have subnets in at least two Availability Zones in a given region. When creating a DB instance in VPC, you must select a DB subnet group. Amazon RDS uses that DB subnet group and your preferred Availability Zone to select a subnet and an IP address within that subnet to associate with your DB instance. If the primary DB instance of a Multi-AZ deployment fails, Amazon RDS can promote the corresponding standby and subsequently create a new standby using an IP address of the subnet in one of the other Availability Zones.

When Amazon RDS creates a DB instance in a VPC, it assigns a network interface to your DB instance by using an IP address selected from your DB subnet group. However, we strongly recommend that you use the DNS name to connect to your DB instance because the underlying IP address can change during failover.

> **Note**
> For each DB instance that you run in a VPC, you should reserve at least one address in each subnet in the DB subnet group for use by Amazon RDS for recovery actions.

Hiding a DB Instance in a VPC from the Internet

One common Amazon RDS scenario is to have a VPC in which you have an EC2 instance with a public-facing web application and a DB instance with a database that is not publicly accessible. For example, you can create a VPC that has a public subnet and a private subnet. Amazon EC2 instances that function as web servers can be deployed in the public subnet, and the Amazon RDS DB instances are deployed in the private subnet. In such a deployment, only the web servers have access to the DB instances. For an illustration of this scenario, see A DB Instance in a VPC Accessed by an EC2 Instance in the Same VPC.

When you launch a DB instance inside a VPC, you can designate whether the DB instance you create has a DNS that resolves to a public IP address by using the PubliclyAccessible parameter. This parameter lets you designate

whether there is public access to the DB instance. Note that access to the DB instance is ultimately controlled by the security group it uses, and that public access is not permitted if the security group assigned to the DB instance does not permit it.

You can modify a DB instance to turn on or off public accessibility by modifying the PubliclyAccessible parameter. This parameter is modified just like any other DB instance parameter. For more information, see the modifying section for your DB engine.

The following illustration shows the Publicly Accessible option in the Launch DB Instance Wizard.

Creating a DB Instance in a VPC

The following procedures help you create a DB instance in a VPC. If your account has a default VPC, you can begin with step 3 because the VPC and DB subnet group have already been created for you. If your AWS account doesn't have a default VPC, or if you want to create an additional VPC, you can create a new VPC. If you don't know if you have a default VPC, see Determining Whether You Are Using the EC2-VPC or EC2-Classic Platform.

> **Note**
>
> If you want your DB instance in the VPC to be publicly accessible, you must update the DNS information for the VPC by enabling the VPC attributes DNS hostnames and DNS resolution. For information about updating the DNS information for a VPC instance, see Updating DNS Support for Your VPC.
>
> Follow these steps to create a DB instance in a VPC:
>
> Step 1: Create a VPC
> Step 2: Add Subnets to the VPC
> Step 3: Create a DB Subnet Group
> Step 4: Create a VPC Security Group
> Step 5: Create a DB Instance in the VPC

Step 1: Create a VPC

If your AWS account does not have a default VPC or if you want to create an additional VPC, follow the instructions for creating a new VPC. See Create a VPC with Private and Public Subnets in the Amazon RDS documentation, or see Step 1: Create a VPC in the Amazon VPC documentation.

Step 2: Add Subnets to the VPC

Once you have created a VPC, you need to create subnets in at least two Availability Zones. You use these subnets when you create a DB subnet group. Note that if you have a default VPC, a subnet is automatically created for you in each Availability Zone in the region.

For instructions on how to create subnets in a VPC, see Create a VPC with Private and Public Subnets in the Amazon RDS documentation.

Step 3: Create a DB Subnet Group

A DB subnet group is a collection of subnets (typically private) that you create for a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when you create DB instances using the CLI or API. If you use the Amazon RDS console, you can just select the VPC and subnets you want to use. Each DB subnet group must have at least one subnet in at least two Availability Zones in the region.

> **Note**
>
> For a DB instance to be publicly accessible, the subnets in the DB subnet group must have an Internet gateway. For more information about Internet gateways for subnets, go to Internet Gateways in the Amazon VPC documentation.

When you create a DB instance in a VPC, you must select a DB subnet group. Amazon RDS then uses that DB subnet group and your preferred Availability Zone to select a subnet and an IP address within that subnet. Amazon RDS creates and associates an Elastic Network Interface to your DB instance with that IP address. For Multi-AZ deployments, defining a subnet for two or more Availability Zones in a region allows Amazon RDS to create a new standby in another Availability Zone should the need arise. You need to do this even for Single-AZ deployments, just in case you want to convert them to Multi-AZ deployments at some point.

Step 4: Create a VPC Security Group

Before you create your DB instance, you must create a VPC security group to associate with your DB instance. For instructions on how to create a security group for your DB instance, see Create a VPC Security Group for a Private Amazon RDS DB Instance in the Amazon RDS documentation, or see Security Groups for Your VPC in the Amazon VPC documentation.

Step 5: Create a DB Instance in the VPC

In this step, you create a DB instance and use the VPC name, the DB subnet group, and the VPC security group you created in the previous steps.

> **Note**
> If you want your DB instance in the VPC to be publicly accessible, you must enable the VPC attributes DNS hostnames and DNS resolution. For information on updating the DNS information for a VPC instance, see Updating DNS Support for Your VPC.

For details on how to create a DB instance for your DB engine, see the topic following that discusses your DB engine. For each engine, when prompted in the Launch DB Instance Wizard, enter the VPC name, the DB subnet group, and the VPC security group you created in the previous steps.

Database Engine	Relevant Documentation
Amazon Aurora	Creating an Amazon Aurora DB Cluster
MariaDB	Creating a DB Instance Running the MariaDB Database Engine
Microsoft SQL Server	Creating a DB Instance Running the Microsoft SQL Server Database Engine
MySQL	Creating a DB Instance Running the MySQL Database Engine
Oracle	Creating a DB Instance Running the Oracle Database Engine
PostgreSQL	Creating a DB Instance Running the PostgreSQL Database Engine

Updating the VPC for a DB Instance

You can use the AWS Management Console to easily move your DB instance to a different VPC.
For details on how to modify a DB instance for your DB engine, see the topic in the table following that discusses your DB engine. In the Network & Security section of the modify page, shown following, for Subnet Group, enter the new subnet group. The new subnet group must be a subnet group in a new VPC.

Database Engine	Relevant Documentation
MariaDB	Modifying a DB Instance Running the MariaDB Database Engine
Microsoft SQL Server	Modifying a DB Instance Running the Microsoft SQL Server Database Engine
MySQL	Modifying a DB Instance Running the MySQL Database Engine
Oracle	Modifying a DB Instance Running the Oracle Database Engine
PostgreSQL	Modifying a DB Instance Running the PostgreSQL Database Engine

Note

Updating VPCs is not currently supported for Aurora clusters.

WordPress

About WordPress

URL Link https://wordpress.org/about/

WordPress started in 2003 with a single bit of code to enhance the typography of everyday writing and with fewer users than you can count on your fingers and toes. Since then it has grown to be the largest self-hosted blogging tool in the world, used on millions of sites and seen by tens of millions of people every day. Everything you see here, from the documentation to the code itself, was created **by and for the community**. WordPress is an Open Source project, which means there are hundreds of people all over the world working on it. (More than most commercial platforms.) It also means you are free to use it for anything from your recipe site to a Fortune 500 web site without paying anyone a license fee and a number of other important freedoms.

About WordPress.org
On this site you can download and install a software script called WordPress. To do this you need a web host who meets the minimum requirements and a little time. WordPress is completely customizable and can be used for almost anything. There is also a **service** called WordPress.com which lets you get started with a new and free WordPress-based blog in seconds, but varies in several ways and is less flexible than the WordPress you download and install yourself.

What You Can Use WordPress For
WordPress started as just a blogging system, but has evolved to be used as full content management system and so much more through the thousands of plugins and widgets and themes, WordPress is limited only by your imagination. (And tech chops.)

Connect with the Community
In addition to online resources like the forums and mailing lists a great way to get involved with WordPress is to attend or volunteer at a WordCamp, which are free or low-cost events that happen all around the world to gather and educate WordPress users, organized by WordPress users. Check out the website, there might be a WordCamp near you.

A Little History
WordPress was born out of a desire for an elegant, well-architectured personal publishing system built on PHP and MySQL and licensed under the GPLv2 (or later). It is the official successor of b2/cafelog. WordPress is fresh software, but its roots and development go back to 2001. It is a mature and stable product. We hope by focusing on user experience and web standards we can create a tool different from anything else out there.

For a bit more about WordPress' history check out the WordPress Wikipedia page or this page on our own Codex.

Writing Posts

URL Link https://codex.wordpress.org/Writing_Posts

Posts are entries that display in reverse order on your home page. Posts usually have comments fields beneath them and are included in your site's RSS feed.

To write a post:
1. Log in to your WordPress Administration Panel (Dashboard).
2. Click the 'Posts' tab.
3. Click the 'Add New' sub-tab.

4. Start filling in the blanks: enter your post title in the upper field, and enter your post body content in the main post editing box below it.
5. As needed, select a category, add tags, and make other selections from the sections below the post. (Each of these sections is explained below.)
6. When you are ready, click **Publish**.

Descriptions of Post Fields

Title/Headline Box

The title of your post. You can use any phrases, words or characters. Avoid using the same title twice as that will cause problems. You can use commas, apostrophes, quotes, hyphens/dashes and other typical symbols in the post like "My Site - Here's Lookin' at You, Kid". WordPress will then clean it up to generate a user-friendly and URL-valid name of the post (also called the "post slug") to compose the permalink for the post.

Body Copy Box

The blank box where you enter your writing, links, links to images, and any information you want to display on your site. You can use either the Visual or the Text view to compose your posts. For more on the Text view, see the section below, Visual Versus Text View.

Preview button

Allows you to view the post before officially publishing it.

Save

Allows you to save your post as a draft / pending review rather than immediately publishing it. To return to your drafts later, visit Posts - Edit in the menu bar, then select your post from the list.

Publish

Publishes your post on the site. You can edit the time when the post is published by clicking the Edit link above the "Publish" button and specifying the time you want the post to be published. By default, at the time the post is first auto-saved, that will be the date and time of the post within the database.

Password Protect This Post

To password protect a post, click Edit next to Visibility in the Publish area to the top right, then click Password Protected, click Ok, and enter a password. Then click OK. Note - Editor and Admin users can see password protected or private posts in the edit view without knowing the password.

Post Author

A list of all blog authors you can select from to attribute as the post author. This section only shows if you have multiple users with authoring rights in your blog. To view your list of users, see Users tab on the far right. For more information, see Users and Authors.

Revisions

A list of all revisions made to the current post or page. Clicking on a revision will open a dedicated revision change where you can compare the current version of the post or page with any previous versions. There is also an option to restore any previous versions.

Note: You can set basic options for writing, such as the size of the post box, how smiley tags are converted, and other details by going to Administration Panels > Settings > Writing. See Writing Options SubPanel.

Best Practices For Posting

You can say or show the world anything you like on your WordPress site. Here are some tips you need to know to help you write your posts in WordPress.

Use Paragraphs

No one likes to read writing that never pauses for a line break. To break your writing up into paragraphs, use double spaces between your paragraphs. WordPress will automatically detect these and insert <p> HTML paragraph tags into your writing.

Use Headings

If you are writing long posts, break up the sections by using headings, small titles to highlight a change of subject. In HTML, headings are set by the use of h1, h2, h3, h4, and so on. By default, most WordPress

Themes use the first, second, and sometimes third heading levels within the site. You can use `h4` to set your own headings. You can use the Heading 4 style from your editing dropdown menu using the Visual Editor, or you can enter your headline manually in the Text Editor by typing:
`<h4>Subtitle of Section</h4>`. To style the heading, add it to your `style.css` style sheet file. For more information on styling headings, check out Designing Headings.

Use HTML

You don't have to use HTML when writing your posts. WordPress will automatically add it to your site, but if you do want control over different elements like boxes, headings, and other additional containers or elements, use HTML.

Spell Check and Proof

There are spell check Plugins available, but even those can't check for everything. Some serious writers will write their posts in a text editor with spell check, check all the spelling and proof it thoroughly before copying and pasting into WordPress.

Think before you post

Ranting on blogs is commonplace today, but take a moment and think about what you are writing. Remember, once it is out there, it can be seen by many and crawled by search engines; and taking things back is harder once it is public. Take a moment to read what you've written before hitting the Publish button. When you are ready, share it with the world.

Write about what you like

You've heard this a thousand times before and it sounds too cliched, but it is true. If you force yourself to write something that you don't really enjoy, it will show. Perhaps you might not have a specific theme for writing when you just start, but that's ok. You'll become more focused later. Just enjoy the experience and write what you like.

Write frequently

Write as frequently as you can, but don't let quantity get in the way of quality. Your viewers come for content, not to spend time reading useless stuff.

Don't use too much slang

Not all the readers will be from your part of the world so make sure people can understand easily.

Don't hide your emotions

Tempting as it might be, don't hide your real emotions. After all that is what a blog is about. If you want, you can stay anonymous and voice your feelings on whatever you are passionate about. You might have strong views on various subjects but let your readers know your passion. What is passion worth if you can't even share it? You'll actually love the discussions it can lead to. The discussions will broaden your own thinking and you might end up making some really good friends.

Consider your readers

Perhaps this sounds weird, but consider who needs to know about your blog before you tell them about your new blogging hobby. Will you be able to write freely if you tell them? How much should you let your readers know about you? Is it ok if your boss or girlfriend reads your posts? If you don't want them to read, take anonymity measures accordingly.

Make use of comments

Comments let people share their ideas. Sometimes, they might not be good, but you can ask such people to shut up. Most of the times, they will and if they don't you can delete their comments. Blogging like real life, can be both fun and not so fun at times. Be prepared. Also, give your people a place to contact you in private if they want to write to you.

Worry about blog design later

Blog design matters, but only to an extent. Don't give up on blogging just because the design isn't coming up as you'd like it to be. Sooner or later, you'll get around the design problems with ease. But continue writing. Content is what attracts your readers, not just the look of your blog.

Don't play too safe

Talk about the real you. Readers aren't impressed by how big your house is, which cool club you belong to, or what the weather is in your hometown. Don't be a bore and put a long post on how you fixed the leaking tap in minutes. Readers don't care about braggers, they care about the real you--how you feel, what gets you excited, why you are the person you are. But if achievements are all that you can talk about, you will bore your readers.

Use pictures and videos

They make the pages colorful and viewers get to see a little of your part of the world. They feel connected.

Keep writing

Don't stop blogging. If you don't have anything to write about, chances are, you are still holding back. Let loose. Perhaps surf more blogs and maybe you'll get an idea. You can write about your friends, complain about your boss, or simply rant about what's gone wrong. Yet if nothing else works, just write a review on the latest movie, book, or product. Easy actually.

Save your posts

Save your posts before you press the publish button. Anything can happen with your computer or with an internet connection. You don't need to lose your post.

Using Themes

URL Link https://codex.wordpress.org/Using_Themes

What is a Theme?

Fundamentally, the WordPress Theme system is a way to "skin" your weblog. Yet, it is more than just a "skin." Skinning your site implies that only the design is changed. WordPress Themes can provide much more control over the look *and presentation* of the material on your website.

A WordPress Theme is a collection of files that work together to produce a graphical interface with an underlying unifying design for a weblog. These files are called **template files**. A Theme modifies the way the site is displayed, without modifying the underlying software. Themes may include customized template files, image files (**.jpg*, **.gif*), style sheets (**.css*), custom Pages, as well as any necessary code files (**.php*). For an introduction to template files, see Stepping Into Templates.

Let's say you write a lot about cheese and gadgets. Through the use of the WordPress Loop and template files, you can customize your Cheese category posts to look different from your Gadgets category posts. With this powerful control over what different pages and categories look like on your site, you are limited only by your imagination. For information on how to use different Themes for different categories or posts, see The Loop in Action and Category Templates.

Get New Themes

The WordPress Theme Directory is the official site for WordPress Themes which have been checked and inspected, and are free for downloading. The site features the ability to search by type and style, and offers a demonstration of the page view elements of the Theme.

Using Themes

WordPress currently comes with three themes: the default Twenty Fifteen theme, and previous defaults Twenty Fourteen theme and Twenty Thirteen theme. You can switch between Themes using the Appearance admin panel. Themes that you add to the theme directory will appear in the Administration Screen > Appearance > Themes as additional selections.

Adding New Themes

There are many Themes available for download that will work with your WordPress installation.

If the Theme that you are installing provides instructions, be sure to read through and follow those instructions for the successful installation of the Theme. **It is recommended that Theme developers provide installation instructions for their own Themes**, because Themes can provide special optional functionality that may require more steps than the basic installation steps covered here. If your Theme does not work after following any provided instructions, please **contact the Theme author for help**.

Adding New Themes using the Administration Panels

You can download Themes directly to your blog by using the Add New Themes option in the Appearance sub-menu.

1. Log in to the WordPress Administration Panels.
2. Select the Appearance panel, then Themes.
3. Select Add New.
4. Either use the Search or Filter options to locate a Theme you would like to use.
5. Click on the Preview link to preview the Theme or the Install Now link to upload the Theme to your blog,
6. Or use the Upload link in the top links row to upload a zipped copy of a Theme that you have previously downloaded to your machine.

Adding New Themes by using cPanel

If your host offers the cPanel control panel, and the Theme files are in a *.zip* or *.gz* archive follow these instructions. Note: This assumes the Theme you download is a compressed (*.zip*) file containing a folder under which all the Theme files reside.

1. Download the Theme *.zip* file to your local machine.
2. In cPanel File Manager, navigate to your Themes folder. If your WordPress is installed in the document root folder of your web server you would navigate to "public_html/wp-content/themes" and if you have WordPress installed in a sub-folder called wordpress, you would navigate to "public_html/wordpress/wp-content/themes".
3. Once you've navigated to the Themes folder in cPanel File Manager, click on Upload file(s) and upload that *.zip* file you saved in Step 1.
4. Once the *.zip* file is uploaded, click on the name of that file in cPanel, then in the panel to the right, click on "Extract File Contents", and that *.zip* file will be uncompressed.
5. Follow the instructions below for selecting the new Theme.

Adding New Themes Manually (FTP)

To add a new Theme to your WordPress installation, follow these basic steps:

1. Download the Theme archive and extract the files it contains. You may need to preserve the directory structure in the archive when extracting these files. Follow the guidelines provided by your Theme author.
2. Using an FTP client to access your host web server, create a directory to save your Theme in the wp-content/themes directory provided by WordPress. For example, a Theme named **Test** should be in wp-content/themes/test. Your Theme may provide this directory as part of the archive.
3. Upload the Theme files to the new directory on your host server.
4. Follow the instructions below for selecting the new Theme.

Selecting the Active Theme

To select a Theme for your site:

1. Log in to the WordPress Administration Panels.
2. Select the Appearance panel, then Themes.
3. From the Themes panel, roll over the Theme thumbnail image for the Theme you are interested in to see options for that theme.
4. You can view more information about any theme by clicking Theme Details.
5. A live preview of any Theme (using your blog's content) can be seen by clicking Live Preview.

6. To activate the Theme click the **Activate** button.

Your selection will immediately become active.
Note: If the Theme preview is blank, do **not** activate the new Theme without investigating further. Your site may not be displayed correctly, otherwise.

Creating Themes

If you are interested in creating your own Theme for distribution, or learning more about the architecture of Themes, please review the documentation regarding Theme Development.

If you simply want to customize your current Theme for your own use, consider creating a Child Theme.

Categories:
Getting Started
Design and Layout
UI Link

Lesson Summary

Virtual Private Cloud (VPC)

Thinks of VPC as a logical datacenter in AWS
Consists of IGWs (or virtual private gateways), route tables, network ACLs, subnets, security groups

VPC span Availability Zones, but do not span Regions
Can have VPCs in every region
Each subnet is mapped directly to an availability zone, subnets can NOT span Availability Zones
VPC Security Groups, Network ACLs and route tables can span subnets

Default VPC vs Custom VPCs

Default VPC is user friendly, allowing you to immediately deploy instances
When creating AWS account, you will have default VPCs in every region in the world
Done to make it easy to get up and running deploying EC2 instances
All subnets in default VPC have a route out to the internet
Each EC2 instance has both a public and private IP address
Note: Unless you choose not to assign public IP
If you delete the default VPC the only way to get it back is to contact AWS
Note: Do NOT delete default VP

Creating a New VPC, not using the VPC Wizard, creates a Custom VPC.
Custom VPCs have default routes, security groups and Network ACLs
By default, Custom VPCs have no Internet access
You will have to create:

- Create Subnets
- Create Internet Gateway and attach to VPC
- Create Route Tables
- Associate Route Tables to Subnets
- Create Security Groups. Add Inbound rules.

Identity and Access Management (IAM)

IAM Key Concepts

IAM consists of the following:
Users
Groups (A way to aggregate our users and apply policies to them collectively)
Role

IAM is universal. It applies to all regions (Users, Groups, and Roles)
New Users have NO permissions when first created
IAM Users can be given Programmatic and/or AWS Management Console Access

IAM Security Credentials

New Users can be assigned Access Key ID & Secret Access Keys when first created or new credentials created later.

- You can NOT use to login (like an account/password), but you can access AWS resources using APIs and CLI using these credentials.

- This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Security Credential Settings

aws_access_key_id – AWS access key.
aws_secret_access_key – AWS secret key.
region – AWS region.
output – output format (json, text, or table)

IAM Roles

IAM Roles are more secure than Security Credentials

Roles are more secure than storing your access key ID and secret access key on EC2 instances
Roles are easier to manage
Role can only be assigned when the EC2 instance is being provisioned

Elastic Compute Cloud (EC2)

Accessing EC2 Instances Using Stored Credentials

Using EC2 Access Key ID and Secret Access Key to provide access is UNSAFE

Anyone who obtains Access Key ID and Secret Access Key can access resources
Do NOT distribute credentials

Accessing EC2 Instances Using IAM Roles

Recommended Access be provided using IAM Role assigned during EC2 Instance Creation

Assigning an IAM Role to an EC2 Instance, can only be done during instance creation

EC2 Security Groups

All inbound traffic is Blocked
All outbound traffic is Allowed
Changes to Security Groups take effect immediately
You can have any number of EC2 Instances within a security group

EC2 Instance Meta-data

Need to query the instance metadata
Used to get information about an instance (such as public IP)
curl http://169.254.169.254/latest/metadata
get http://169.254.169.254/latest/meta-data
Key thing to remember is that it's an instances META DATA, not user data.

Amazon Web Services Command Line Interface (CLI)

AWS CLI

The AWS Command Line Interface is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

Relational Database Service (RDS)

Database Services Overview

RDS - Managed Relational Database Service

DynamoDB - Managed No-SQL Database Service
ElastiCache - In-memory Cache
Redshift - Fast, Simple, Cost-Effective Data Warehousing
DMS - Managed Database Migration Service

Relational Database Service DB Instance Types

SQL Server
Oracle
MySQL Server
PostgreSQL
Aurora
MariaDB

RDS DB Subnet Groups

Unless you are working with a legacy DB instance, your DB instance is in a virtual private cloud (VPC). The VPC can either be a default VPC that comes with your account or one that you create.

A DB subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when creating DB instances using the CLI or API; if you use the console, you can just select the VPC and subnets you want to use.

Each DB subnet group should have subnets in at least two Availability Zones in a given region. When creating a DB instance in VPC, you must select a DB subnet group.

WordPress

WordPress is well-known for its ease of installation. Under most circumstances, installing WordPress is a very simple process and takes less than five minutes to complete. WordPress is used on 80% of all Websites.