



Immersion Day

*Getting Started with Windows Server on
Amazon EC2*

Table of Contents

Overview.....	3
Launch a Web Server Instance	4
Browse the Web Server.....	12
Connecting To Your Windows Instance.....	13
Appendix – Additional EC2 Concepts.....	16
Change the Instance Type.....	16
Black Belt Booting	18

Overview

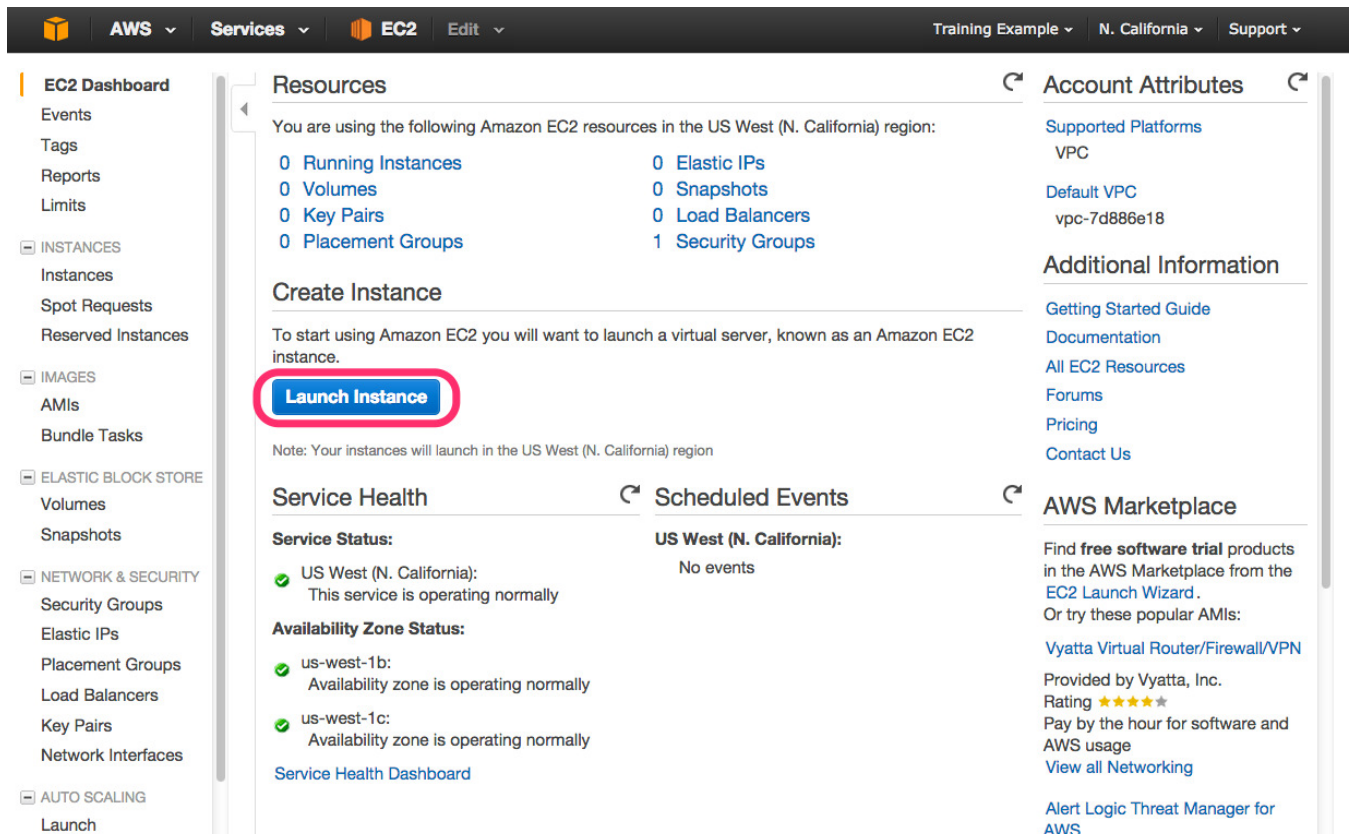
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.

This lab will walk you through launching, configuring, and customizing an EC2 web server using the AWS Management Console.

Launch a Web Server Instance

In this example we will launch a Windows Server 2012 instance with the IIS web server installed upon boot.

1. Sign into the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2>.
2. Click on **Launch Instance**



3. Scroll down and click **Select** on the Windows Server 2012 R2 Base AMI.

Immersion Day

Getting Started with Windows Server on Amazon EC2

Step 1: Choose an Amazon Machine Image (AMI)

[Cancel and Exit](#)


Quick Start

My AMIs

AWS Marketplace


Community AMIs

☐ Free tier only ⓘ

**Amazon Linux**
Free tier eligible


Amazon Linux AMI 2014.09.1 (HVM) - ami-b5a7ea85
The Amazon Linux AMI is an EBS backed image. It includes the 3.14 kernel, Ruby 2.1, PHP 5.5, PostgreSQL 9.3, Docker 1.2, the AWS command line tools, and repository access to many other packages.
Root device type: ebs Virtualization type: hvm

Select
64-bit

**Red Hat**
Free tier eligible


Red Hat Enterprise Linux 7.0 (HVM), SSD Volume Type - ami-99bef1a9
Red Hat Enterprise Linux version 7.0 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm

Select
64-bit

**SUSE Linux**
Free tier eligible


SUSE Linux Enterprise Server 12 (HVM), SSD Volume Type - ami-d7450be7
SUSE Linux Enterprise Server 12 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs Virtualization type: hvm

Select
64-bit

**Ubuntu**
Free tier eligible

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-3d50120d
Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root device type: ebs Virtualization type: hvm

Select
64-bit

**Windows**
Free tier eligible

Microsoft Windows Server 2012 R2 Base - ami-29d18719
Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]
Root device type: ebs Virtualization type: hvm

Select
64-bit

< 1 to 22 of 22 AMIs >

Getting Started with Windows Server on Amazon EC2

4. In the **Choose Instance Type** tab, select the **t2.medium** instance size and click **Next: Configure Instance Details**

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.medium (Variable ECUs, 2 vCPUs, 2.5 GHz, Intel Xeon Family, 4 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
<input type="checkbox"/>	General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High
<input type="checkbox"/>	Compute optimized	c4.large	2	3.75	EBS only	Yes	Moderate
<input type="checkbox"/>	Compute optimized	c4.xlarge	4	7.5	EBS only	Yes	Moderate

[Cancel](#)
[Previous](#)
[Review and Launch](#)
[Next: Configure Instance Details](#)

5. On the **Configure Instance Details** page, expand the **Advanced Details** section, copy/paste [this script](https://immersionday.s3.amazonaws.com/scripts/ec2-windows-lab-userdata.txt) (<https://immersionday.s3.amazonaws.com/scripts/ec2-windows-lab-userdata.txt>) into the User Data field (this script will install/start IIS and deploy a simple web page) and click **Next: Add Storage**:

Getting Started with Windows Server on Amazon EC2

Step 3: Configure Instance Details

Subnet: No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP: Use subnet setting (Enable)

IAM role: None

Shutdown behavior: Stop

Enable termination protection: ☐ Protect against accidental termination

Monitoring: ☐ Enable CloudWatch detailed monitoring
[Additional charges apply.](#)

Tenancy: Shared tenancy (multi-tenant hardware)
[Additional charges will apply for dedicated tenancy.](#)

Advanced Details

User data: ☒ As text ☐ As file ☐ Input is already base64 encoded

```
<powershell>
Install-WindowsFeature Web-Server -IncludeManagementTools -IncludeAllSubFeature
remove-item -recurse c:\inetpub\wwwroot\*
(New-Object System.Net.WebClient).DownloadFile("https://immersionday.s3.amazonaws.com/scripts/ec2-windows.zip",
"c:\inetpub\wwwroot\ec2-windows.zip")

$shell = new-object -com shell.application
$zip = $shell.Namespace("c:\inetpub\wwwroot\ec2-windows.zip")
foreach($item in $zip.items())
{
    $shell.Namespace("c:\inetpub\wwwroot\").copyhere($item)
}
</powershell>
```

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

6. On the **Step 4: Add Storage** screen, Click **Next: Tag Instance** to accept the default Storage Device Configuration and move to the Step 5: Tag Instance screen.

Next, choose a “friendly name” for your instance. This name, more correctly known as a tag, will appear in the console once the instance launches. It makes it easy to keep track of running machines in a complex environment. Name yours according to this format: “[Your Name] Web Server”.

Then click **Next: Configure Security Group**.

Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	John Doe Web Server

[Create Tag](#) (Up to 10 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

Getting Started with Windows Server on Amazon EC2

7. You will be prompted to create a new security group, which will be your firewall rules. On the assumption that we are building out a Web server, name this security group according to this format “[Your Name] Web Server”, and open ports 3389 and 80. Click the **Review and Launch** button after configuring the security group.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source
RDP	TCP	3389	Anywhere 0.0.0.0/0
HTTP	TCP	80	Anywhere 0.0.0.0/0

Add Rule

Warning


Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous **Review and Launch**

8. Review your choices, and then click **Launch**.

Step 7: Review Instance Launch

AMI Details [Edit AMI](#)

 **Microsoft Windows Server 2012 R2 Base - ami-29d18719**
 Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]
 Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.medium	Variable	2	4	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: John Doe Web Server
 Description: launch-wizard-1 created 2015-01-16T17:41:38.720-05:00

Type	Protocol	Port Range	Source
RDP	TCP	3389	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0

Instance Details [Edit instance details](#)

Storage [Edit storage](#)

Cancel Previous **Launch**

9. Now you need to create a public/private keypair. When this instance launches, you will connect to it via Remote Desktop using the credentials for “administrator”. For Windows instances, EC2 automatically generates a password and encrypts with your public key. To decrypt the encrypted password, you will use your private key. Let’s create a new public/private keypair.

Select an existing key pair or create a new key pair


A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. [Learn more about removing existing key pairs from a public AMI.](#)

Choose an existing key pair

Select a key pair

No key pairs found

**No key pairs found**

You don't have any key pairs. Please create a new key pair by selecting the **Create a new key pair** option above to continue.

Cancel Launch Instances

10. Enter a name for the keypair using the following format: [YourName]-KeyPair and click **Download Key Pair**.

Select an existing key pair or create a new key pair ×

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.


Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

JohnDoe-KeyPair

Download Key Pair

 You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel

Launch Instances

11. Your browser will download the private portion of the keypair to your PC. It will have a name like *JohnDoe-KeyPair.pem*. Note the location of this file because you will need it later to decrypt the administrator password.

12. Now click the Launch Instances button to launch your Windows web server.

13.

14. The next screen will confirm that your instance is now launching. Click the **View Instances** button. Once your instance has launched, you will see the “[Your Name] Web Server” instance as well as the Availability Zone the instance is in and its publicly routable DNS name.

Launch Status



Your instances are now launching

The following instance launches have been initiated: [i-0c218402](#) [View launch log](#)



Get notified of estimated charges

[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to your instances.

▼ Here are some helpful resources to get you started

- [How to connect to your Windows instance](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: User Guide](#)
- [Amazon EC2: Microsoft Windows Guide](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

- [Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)
- [Create and attach additional EBS volumes](#) (Additional charges may apply)
- [Manage security groups](#)

[View Instances](#)

Browse the Web Server

1. Wait for the instance to pass the Status Checks. For Windows instances, this could take up to 20 minutes.

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
<input type="checkbox"/>	John Doe Web Server	i-0c218402	t2.medium	us-west-2c	running	Initializing	None

When complete, you will see the Status Checks have passed.

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
<input type="checkbox"/>	John Doe Web Server	i-0c218402	t2.medium	us-west-2c	running	2/2 checks passed	None

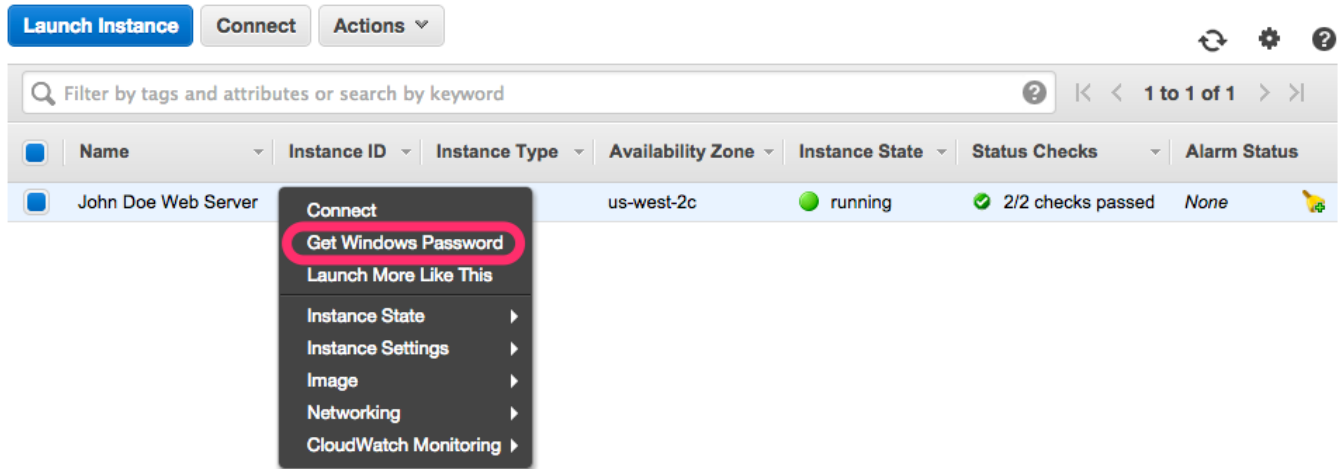
2. Open a new browser tab and browse the Web Server by entering the EC2 instance's Public DNS name into the browser. The EC2 instance's Public DNS name can be found in the console by reviewing the "Public DNS" column. You should see a page that looks similar to this:



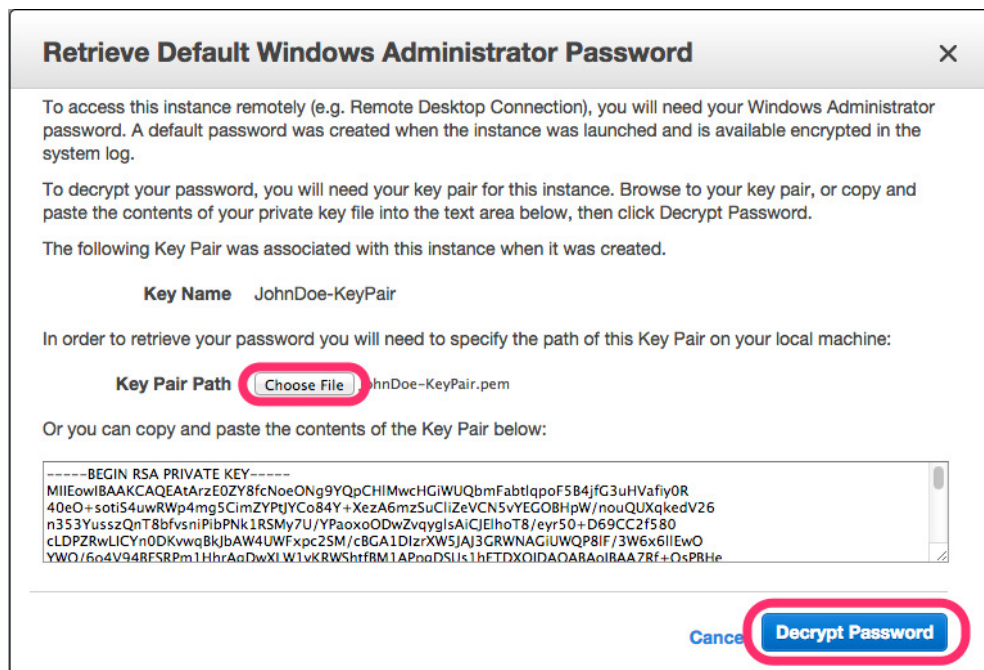
Connecting To Your Windows Instance

To connect to the Windows desktop, we will use a RDP client. If you're using a Windows PC, use the bundled Remote Desktop application. For Mac users, if you don't have a RDP client already installed, download [Microsoft Remote Desktop](#).

1. Retrieve the automatically generated, encrypted Windows password by right clicking your instance and selecting **Get Windows Password**.



2. On the next screen, click the Choose File button and select the private key file that was automatically downloaded earlier when you launched the instance. Then click Decrypt Password to obtain the Administrator password.



3. The decrypted Administrator password should look something like this:



Note that since only you have the private key, it's important to understand the automatically generated password can only be decrypted by you. So it's important to keep this key secure. Generally, the automatically generated password is changed by the customer after first login. If the automatically generated password is not changed and the private key is lost, there's no way to recover the password.

4. Start your RDP application and connect to the hostname of your instance. The hostname can be found in a couple of different places. For example, in the web console, you'll see hostname listed as the **Public DNS** of the instance.

Immersion Day

Getting Started with Windows Server on Amazon EC2

The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below these is a search bar and a table of instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, and Alarm Status. One instance is listed: 'John Doe Web Server' with ID 'i-0c218402', type 't2.medium', in 'us-west-2c' availability zone, and state 'running'. Below the table, the details for the selected instance are shown. The 'Public DNS' field is highlighted with a red box, showing the address 'ec2-54-149-223-97.us-west-2.compute.amazonaws.com'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
John Doe Web Server	i-0c218402	t2.medium	us-west-2c	running	2/2 checks passed	None

Instance: i-0c218402 (John Doe Web Server) Public DNS: ec2-54-149-223-97.us-west-2.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID i-0c218402 Public DNS ec2-54-149-223-97.us-west-2.compute.amazonaws.com

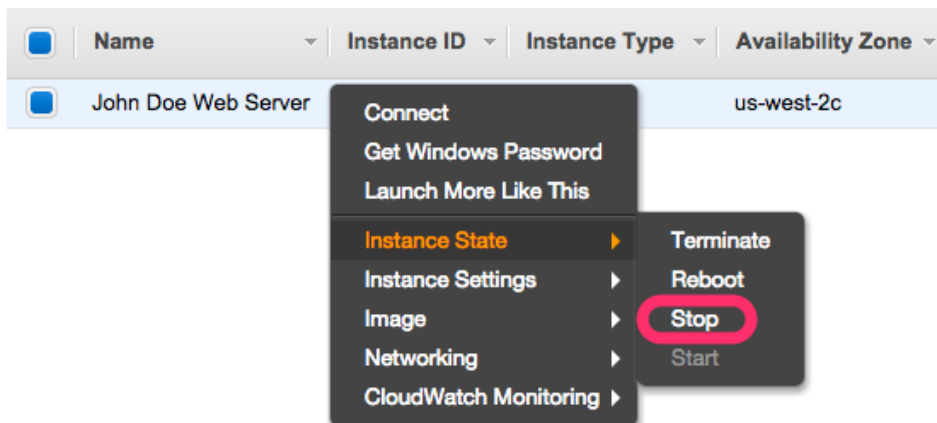
5. In your RDP application, use **Administrator** as the username along with the decrypted password. Once connected, you will have access to the Windows desktop. At this point, feel free to explore Windows. You may want to change the Administrator password to something friendlier or easy to remember (but still secure of course).

Appendix – Additional EC2 Concepts

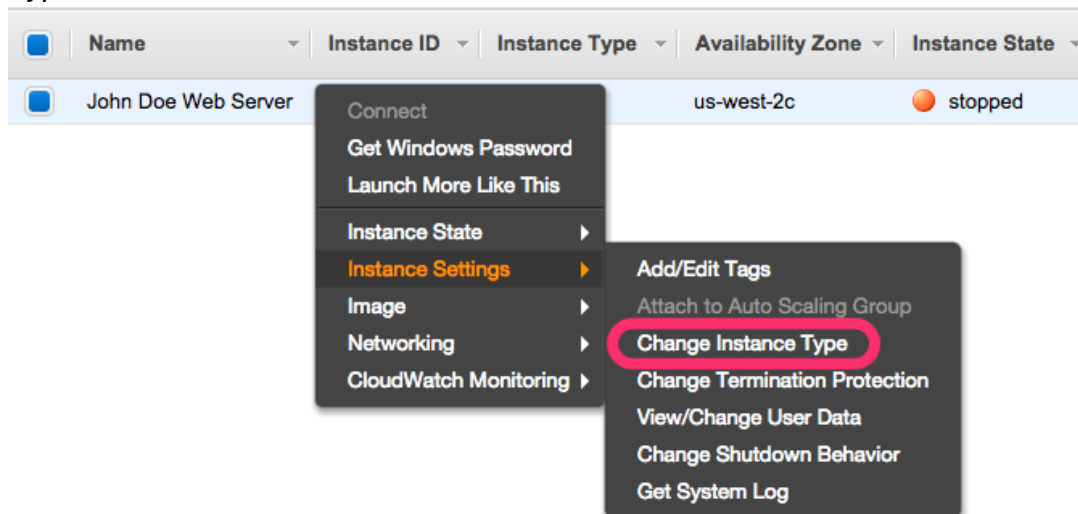
Change the Instance Type

Did you know that you can change the instance type that an AMI is running on? This only works with EBS-backed instances (what we're running here). There is no particular reason to change the instance type in this lab, but the following steps outline how easy it is to do in AWS.

In the AWS Console, select your lab instance, then right-click on it and hover over "Instance State" and select "Stop" (NOT "Terminate"!). Then select "Yes, Stop" to confirm.



After it has stopped, right-click on it again, hover over "Instance Settings" and select "Change Instance Type".



After going through the options and selecting your new instance type, right-click your lab instance and start it again.

Elastic IPs

How do you set up practical DNS names for your web server? Using an address such as <http://ec2-75-101-197-112.compute-1.amazonaws.com/> is cumbersome. Setting up a DNS record that points to <http://www.yourdomain.com> is easy enough – until you reboot the server and the underlying DNS name and IP address both change.

AWS offers Elastic IP Addresses, which are actually NAT addresses that operate at a regional level. That is, an Elastic IP Address works across Availability Zones, within a single region.

Assign one to your application as follows:

- Click on the Elastic IPs link in the AWS Console
- Allocate a new address
- Associate the address with a running instance. If you change instances, it's as simple as allocating the address to the new instance.
- Create a DNS "A" record in your own DNS server that points www.yourdomain.com to 75.101.162.40.

Two Important Notes:



1. As long as an Elastic IP address is associated with a running instance, there is no charge for it. However an address that is not associated with a running instance costs \$0.01/hour. This prevents address hoarding; however it also means that you need to delete any addresses you create, or you will incur an ongoing charge.
2. Load balancing (covered in the next section) requires CNAME records instead of "A" records. So Elastic IP is not required for load-balanced applications.

Black Belt Booting

There are a number of advanced techniques that offer additional power and flexibility when booting instances. For example, some organizations maintain a series of generic instances, and customize the images upon launch.

Common techniques include:

- Automatically check for updates upon each boot.
- Look in a well-known location, such as in a S3 bucket, for data or a script to tell the instance which packages to load.
- Pass **user data** to the instance to accomplish each of the above, or possibly instead of the other approaches.

How to Pass User Data

The general format looks like this from the command line:

```
aws ec2 run-instances --image-id [ami id] --user-data "user data up to  
2048 bytes" ...other params...
```

You can also paste user data into a text field via the AWS Console; however this is usually a form of automation – thus the command line example.