



Automating Windows Workloads

Lab Guide

Prepared by:

Kevin Epstein

Onica Group LLC

Table of Contents

OVERVIEW.....	3
SETTING UP THE VPC	3
LAUNCH A "WORKSTATION".....	4
CREATING THE NEW AMI.....	6
LAUNCH THE WEB APPLICATION STACK.....	8

Overview

This lab will walk you through some of the automation tools AWS provides to help you automate your infrastructure as well as your Windows workloads. The lab uses other tools that are provided by 3rd parties, such as CloudBees for Jenkins, and HashiCorp for Packer. All the tools used in this lab are free for anyone to use.

The code required to run this lab is available publicly on Github.

<https://github.com/kepstein/PackerDemo>
<https://github.com/kepstein/WindowsDemo>

There is also an initial CloudFormation template that you need to run. That template is available at:

<https://s3-us-west-2.amazonaws.com/corpinfo/WindowsLABv2.json>

Setting up the VPC

In this section, you will create a new VPC in which the rest of the lab will be built. The VPC will be created with a CloudFormation template. CloudFormation allows you to express your infrastructure as code, meaning you can now treat your infrastructure as you would any other piece of code, including checking it into some sort of version control, or setting up automated testing of your infrastructure.

1. Ensure you have switched your AWS console to the US-WEST-2 region (Oregon)
2. Navigate to CloudFormation on the AWS Console
3. Click the "Create Stack" button
4. On the "Select Template" screen, choose the option "Specify an Amazon S3 template URL". Copy the URL below, and paste it into the form field, and then click "Next"

<https://s3-us-west-2.amazonaws.com/corpinfo/WindowsLABv2.json>

5. Give the stack a name. For this Lab use something like "WindowsLab"
6. In the "Parameters" section you will see a parameter called "SourceIP". You need to get your current external IP address, and paste the IP address into the form field, then click "Next"
7. On the "Options" page, leave everything as default and click "Next"
8. On the "Review" page, review your settings, and check the "I acknowledge that this template might cause AWS CloudFormation to create IAM resources." Box near the bottom of the screen, and then click "Create"

9. CloudFormation will start creating your Stack, and you should see your stack in the "CREATE_IN_PROGRESS" status. If there are no errors, after a couple minutes you should see your stack with the "CREATE_COMPLETE" status.
10. When you see the "CREATE_COMPLETE" status, you should review the "Outputs" tab. There will be important and useful information that you will use later in this lab.
11. To simplify your workflow, it might be helpful to copy the output to a text editor and save the text file.

Create StackActions ▾Design template

Filter: Active ▾By Name: Windows✕

	Stack Name	Created Time	Status	Description
✓	WindowsLab	2016-06-15 22:06:51 UTC-0700	CREATE_COMPLETE	VPC Environments for Windows Automation Lab

OverviewOutputsResourcesEventsTemplateParametersTagsStack PolicyChange Sets

2016-06-15	Status	Type	Logical ID	Status
▶ 22:10:13 UTC-0700	CREATE_COMPLETE	AWS::CloudFormation::Stack	WindowsLab	Resou
▶ 22:10:11 UTC-0700	CREATE_COMPLETE	AWS::IAM::InstanceProfile	WindowsDemoInstanceProfile	
▶ 22:08:10 UTC-0700	CREATE_IN_PROGRESS	AWS::IAM::InstanceProfile	WindowsDemoInstanceProfile	
22:08:10 UTC-0700	CREATE_IN_PROGRESS	AWS::IAM::InstanceProfile	WindowsDemoInstanceProfile	
▶ 22:08:07 UTC-0700	CREATE_COMPLETE	AWS::IAM::Role	WindowsDemoRole	
▶ 22:08:06 UTC-0700	CREATE_COMPLETE	AWS::IAM::Role	CodeDeployRole	

Launch a "Workstation"

In this section, we will launch a Windows Server instance. This instance has Jenkins, Packer, Git and a couple other supporting pieces of software that will be used for the remainder of the lab.

1. Navigate to EC2 from the AWS Console.
2. Click the blue "Launch Instance" button, on the left side of the screen, click the "Community AMIs" link and search for "CorplInfo". Select the AMI that says "Workstation", and launch the instance, by clicking the "Select" button.
 - a. Your first step is to select an instance type and size. We recommend a t2.medium. Select a t2.medium instance, and then click "Next: Configure Instance Details"
 - b. Network : Select the VPC named "[WindowsLab](#)"
 - c. Subnet: [Select any one of the two subnets](#)
 - d. Auto-assign Public IP: [Enable](#)
 - e. IAM Role: [WindowsLab-DemoInstanceProfile-XXXXXXXXXX](#)

Step 3: Configure Instance Details

Number of instances ⓘ 1 [Launch into Auto Scaling Group ⓘ](#)

Purchasing option ⓘ ☐ Request Spot instances

Network ⓘ vpc-cd52feaa (10.0.0.0/16) | WindowsLab [↕](#) [↻](#) [Create new VPC](#)

Subnet ⓘ subnet-a1782ff9(10.0.1.0/24) | app_subnet_1 | us-east-1a [↕](#) [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP ⓘ Enable [↕](#)

Domain join directory ⓘ None [↕](#) [↻](#) [Create new directory](#)

IAM role ⓘ WindowsLab-WindowsDemoInstanceProfile-1CLATRHEGOJG [↕](#) [↻](#) [Create new IAM role](#)




Shutdown behavior ⓘ Stop [↕](#)

- f. Click "Next: Add Storage"
- g. Accept the defaults for disk space, and click "Next: Add Tags"
- h. Give the instance the name "Workstation", and click "Next: Configure Security Group"
- i. On the "Configure Security Group" page, select the option "Select an [existing](#) security group", and select the security groups that looks like "[WindowsLab-WebServerSecurityGroup-XXXXXXXXXX](#)", and then click the "Review and Launch" button

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

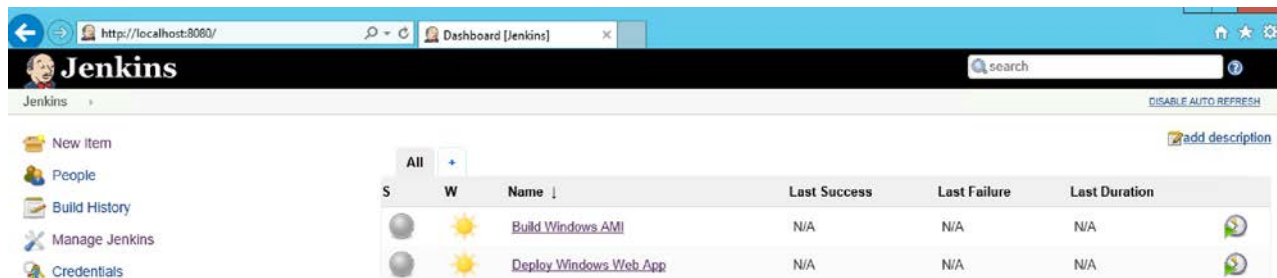
Security Group ID	Name	Description	Actions
 sg-48ea8b33	default	default VPC security group	Copy to new
 sg-1bea8b60	WindowsLab-PackerSecurityGroup-4GMJEC9KO1DQ	Enable RDP and WINRM for Packer	Copy to new
 sg-25ea8b5e	WindowsLab-WebServerSecurityGroup-ODA9RFOPHJV9	Enable RDP and HTTP for Windows EC2 Servers	Copy to new

Inbound rules for sg-25ea8b5e (Selected security groups: sg-25ea8b5e)

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
HTTP	TCP	80	0.0.0.0/0
RDP	TCP	3389	198.11.218.38/32

- j. Review your configurations and then click the "Launch" button.
- k. You will be prompted to select an existing key pair, or create a new one. For this lab, this step is not important since the instance already has a customized password, and you will not need your key pair to retrieve the Windows password as you normally would. Go ahead and either create a new key pair, or select an existing one, or choose the option "Proceed without a key pair", check the acknowledgement box, and then click "Launch Instances"

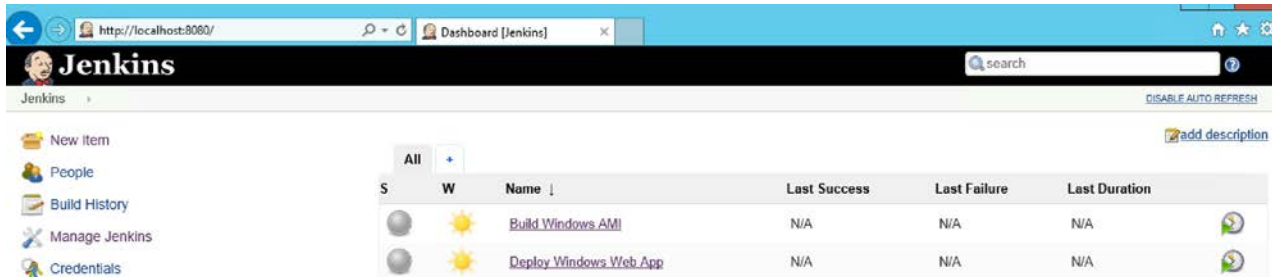
3. Connect to your workstation instance. You can find the Public IP for you instance in the EC2 console. Use the username "[Administrator](#)" and the password "[ChangeMeNow!123](#)"
4. Verify that you are able to browse to <http://localhost:8080>. You should see the Jenkins web application load. There should be two Jenkins tasks.
 - a. The first task, "Build Windows AMI", will use Packer to create a new Amazon Machine Image (AMI). Packer will install, AWS CodeDeploy, Chef, and IIS.
 - b. The second task, "Deploy Windows Web App", will use the AMI that is created by the first task. This task will use CloudFormation to create an "Application Stack". CloudFormation will create the required Security Groups, IAM roles, and Instance profiles that the application stack will need. CloudFormation will also create an Auto Scaling group, and launch configuration. It will also create a new "Application" in AWS CodeDeploy, and deploy the new application to the Auto Scaling instance (there should be 2) that started automatically as a result of deploying this application stack with CloudFormation.



Creating the new AMI

In this section of the lab you will use Jenkins to run a Packer job to create a new AMI. The AMI will only get some very simple updates, but you could use this approach to do much more sophisticated configurations on your Windows Servers in your environment. This process is often called "baking an AMI", since you pre- install and configure as much as you possibly can leave the bare minimum amount of work left for when the instance starts. This means that in production, when you Auto Scale your application, new instances will be ready much more quickly to perform production work. In this example, the only thing missing from the AMI that will be created, is the actual code for the page. For this stage, we will be using Jenkins as an orchestrator, and Packer as the tool to create the AMI. Packer will leverage some PowerShell scripts to make the required configurations to a machine, that will ultimately be converted into an AMI.

1. From the Jenkins default page, where you see the two jobs, click the build icon for the "Build Windows AML" job.



2. You will need to reconfigure the job before you can run it.
 - a. Go to IAM and generate ACCESS and SECRET keys, if you don't already have some.
 - i. These are created under username (top right of console) -> Security Credentials
 - b. Hover the mouse over the Job name "Build Windows AML". You should see a small arrow pointing down, on the right of the job name. Click that arrow, and then click "Configure"
 - c. Scroll down until you find "Build Environment". Below that you should find "Bindings". Click the "Add" button, and enter your ACCESS and SECRET keys. Leave the ID field empty, but do enter a description.

Build Environment

- ☐ Create AWS Cloud Formation stack
- ☒ Use secret text(s) or file(s)

Bindings

Username and password (separated)

Username Variable

Password Variable

Credentials

☒ Specific credentials ☐ Parameter expression



Add

- d. Once your keys are entered, make you selection your keys from the dropdown.
 - e. Save the task.
3. Click "Build with Parameters" (found on the left).
4. You will need to enter some details into the parameters. These details can be obtained from the "Outputs" from the original CloudFormation stack you created. Fill in all the required information and click "Build"

Project Build Windows AMI

This build requires parameters:

PACKER_SG	sg-77e6860c
	The Security group for Packer. Refer to the "PackerSG" value in the outputs from the VPC you created with CloudFormation.
PACKER_SUBNET	subnet-efa8ffb7
	Select one of the subnet id's from the "WebAppSubnets" value in the outputs section of the CloudFormation you previously launched
INSTANCE_PROFILE	WindowsLab-WindowsDemoInstanceProfile-145YANFFXYMZU
	Search IAM for the Role with a name that begins with "WindowsLab-WindowsDemoRole-", and past that role name in this parameter.
Build	

- The task will take some time to complete. You can watch the progress of the job by clicking build number, and then on the "Console Output" link. You should see the status of your job, and when it's completed successfully, the end of the output will look something like the screenshot below.

```
==> amazon-ebs: Creating the AMI: Windows-Demo-AMI-2
    amazon-ebs: AMI: ami-226cab4f
==> amazon-ebs: Waiting for AMI to become ready...
==> amazon-ebs: Adding tags to AMI (ami-226cab4f)...
    amazon-ebs: Adding tag: "Name": "Windows Demo"
==> amazon-ebs: Tagging snapshot: snap-e41ele00
==> amazon-ebs: Terminating the source AWS instance...
==> amazon-ebs: Cleaning up any extra volumes...
==> amazon-ebs: No volumes to clean up, skipping
==> amazon-ebs: Deleting temporary keypair...
Build 'amazon-ebs' finished.

==> Builds finished. The artifacts of successful builds are:
--> amazon-ebs: AMIs were created:

us-east-1: ami-226cab4f
Finished: SUCCESS
Finished: SUCCESS
```

Launch the Web Application Stack

- From the Jenkins default page click the "Deploy Windows Web App" build icon in the same way that you did for the Packer job.
- You will again, have to edit the job, and select the credentials in the same way that you did for the Build AMI task.
- Once you have updated the credentials, save the task, and run it with the "Build with Parameters" link.
- Fill in all the required parameters, and click the build button.

Project Deploy Windows Web App

This build requires parameters:

SUBNETS	<input type="text" value="subnet-efa8ffb7,subnet-4f98d265"/>
Replace these values with the correct values from your Windows Lab VPC	
LAUNCH_AMI	<input type="text" value="ami-226cab4f"/> x
The AMI ID for the Launch Config	
IP	<input type="text" value="198.11.218.38"/>
Your public IP address	
VPC_ID	<input type="text" value="vpc-50cc6337"/>
The VPC ID from the Windows Lab	
SERVICE_ARN	<input type="text" value="arn:aws:iam::730621689362:role/WindowsLab-CodeDeployRole-1VVEHEUB1HTEY"/>
The CodeDeploy Service ARN	
IAM_PROFILE	<input type="text" value="arn:aws:iam::730621689362:instance-profile/WindowsLab-WindowsDemoInstanceProfile-145YANFFXYMZU"/>
The IAM Instance profile for the auto scaling instance	

Build

5. This task will take a while to run. As you did before, you can watch the task progress from the "Console Output". You can also navigate to CloudFormation on the AWS Console, and watch the progress of your stack there.