



Securing your DevOps Toolchain in the Cloud



Cisco Public 2017 All Rights Reserved

Tony Rice
Cisco Security & Trust Organization

Prerequisites

If you don't already have these established,
please take care of this now

1. Amazon AWS account (free tier is fine)
 - aws.amazon.com/free
2. Github.com account



Python Prep (for local testing)

- Install Python

<https://www.python.org/downloads/>

- Install PIP

<https://pip.pypa.io>

`pip install <module>`

- `virtualenv`

- `pip install flask==0.10.1`

About Me

Senior Information Security Engineer

Cisco Security & Trust Organization

Research Triangle Park, North Carolina



| Certified Information
Systems Security Professional



In this workshop we will

- Establish a source code repository (github)
- Bring up a centralized build server (Jenkins)
- Deploy a simple micro service into the serverless cloud (Elastic Beanstalk)
- Form them into a continuous integration toolchain ... *securely by storing credentials*

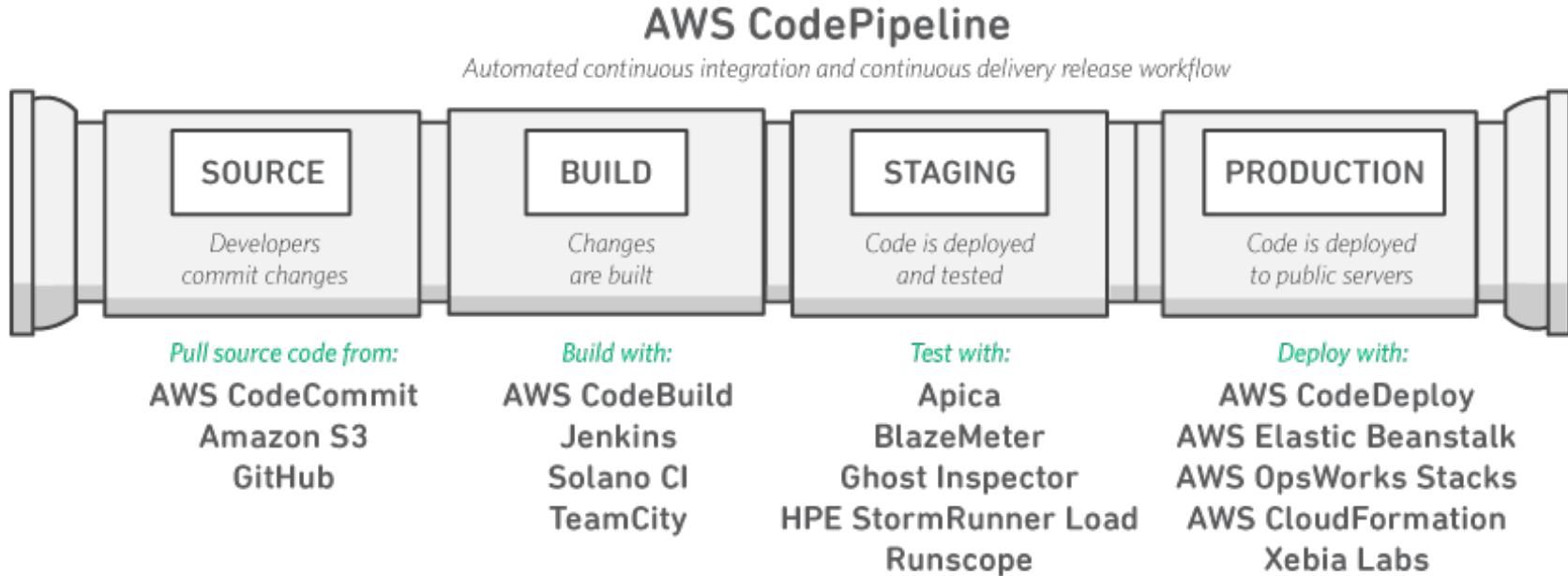
In this workshop is not

- An endorsement of GitHub, Jenkins, any AWS service or anything else used here. These are convenient examples, nothing more, nothing less.
- Leave anyone behind... at least we'll try our hardest

Agenda

- 1-1:30 overview / AWS and Git prerequisites
- 1:30-1:45 Fork the GitHub Hello Word Repo
- 1:45-2:30 Provision EC2, Elastic Beanstalk and security groups
- 2:30-3:00 install Jenkins with Git and EB plugins
- break
- 3:30-4:30
 - trigger Jenkins builds from GitHub updates via keys,
 - trigger deployment from Jenkins to Elastic Beanstalk via keys
- 4:30-5:00 CI/CD best practices

Continuous Integration / Continuous Delivery



Get the app source code from GitHub

- Visit <https://github.com/rtpphokie/hello-world>

Pull ...
then clone

rtphokie / hello-world

No description, website, or topics provided.

4 commits 1 branch 0 releases 1 contributor GPL-3.0

New pull request Create new file Upload files Find file Clone or download

File	Description	Time
.gitignore	Initial commit	2 hours ago
LICENSE	Initial commit	2 hours ago
README.md	updated README	4 minutes ago
application.py	converted to a microservice doing something useful	5 minutes ago
ncaa.dat	converted to a microservice doing something useful	5 minutes ago

Clone

Get the app running locally

1. `virtualenv ~/eb-virt`
2. `source ~/eb-virt/bin/activate`
3. `git https://github.com/rtpphokie/hello-world.git`
or download from Box (<http://bit.ly/isaca-hello-world>)
4. `cd hello-world`
5. `pip install -r requirements`

Test application locally

1. `python application.py`
2. Visit `http://localhost:5000`

Hello NCAA

This is a RESTful micro service returning information on NCAA Div I football teams

- *param/search string*
where *param* is school, conference, mascot, city or state and

example: `http://localhost:5000/school/bama`



AWS: security group

Create Security Group

Security group name: Jenkins 8080

Description: Jenkins 8080

VPC: vpc-4ac6d728 (default)

Security group rules:

Inbound Outbound

Type	Protocol	Port Range	Source	Description
Custom TCP	TCP	8080	Anywhere	0.0.0.0/0, ::/0
Jenkins				

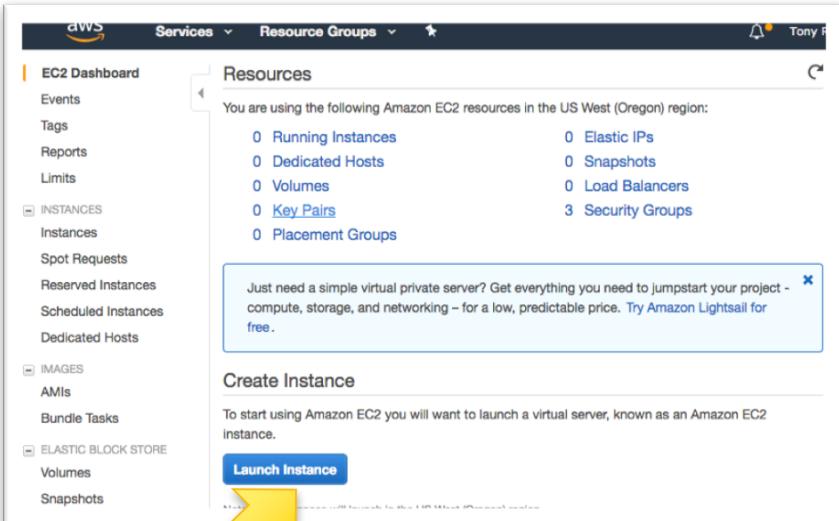
Add Rule

 Create

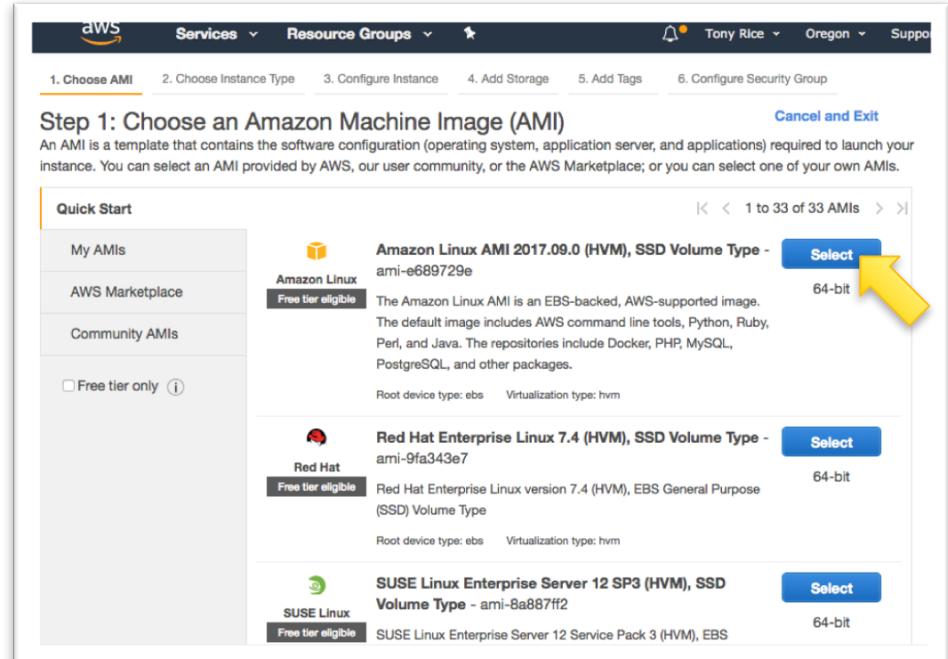
AWS: launch EC2 instance

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with sub-links: Instances, Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts), IMAGES (with sub-links: AMIs, Bundle Tasks), and ELASTIC BLOCK STORE (with sub-links: Snapshots). A large yellow arrow points from the bottom-left towards the 'Launch Instance' button. The main content area is titled 'Resources' and displays the following information: You are using the following Amazon EC2 resources in the US West (Oregon) region: 0 Running Instances, 0 Dedicated Hosts, 0 Volumes, 0 Key Pairs, 0 Placement Groups, 0 Elastic IPs, 0 Snapshots, 0 Load Balancers, and 3 Security Groups. Below this, there's a promotional message: "Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking – for a low, predictable price. Try Amazon Lightsail for free." At the bottom, there's a section titled 'Create Instance' with the text: "To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance." A prominent blue 'Launch Instance' button is centered at the bottom of this section. A note at the very bottom states: "Note: Your instances will launch in the US West (Oregon) region".

AWS: launch EC2 instance



The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with various navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts, AMIs, Bundle Tasks, and Elastic Block Store. The main area is titled 'Resources' and shows a summary of resources in the US West (Oregon) region: 0 Running Instances, 0 Dedicated Hosts, 0 Volumes, 0 Key Pairs, 0 Placement Groups, 0 Elastic IPs, 0 Snapshots, 0 Load Balancers, and 3 Security Groups. Below this is a callout box with text: 'Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking – for a low, predictable price. Try Amazon Lightsail for free.' At the bottom, there's a 'Create Instance' section with a 'Launch Instance' button.



The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' page. It has a header with tabs: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, and 6. Configure Security Group. The '1. Choose AMI' tab is selected. The page is titled 'Step 1: Choose an Amazon Machine Image (AMI)'. It says: 'An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.' There's a 'Quick Start' sidebar with 'My AMIs', 'AWS Marketplace', and 'Community AMIs'. Below is a list of AMIs: 1. Amazon Linux AMI 2017.09.0 (HVM), SSD Volume Type - ami-e689729e (Selected, 64-bit). 2. Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-9fa343e7 (Free tier eligible, 64-bit). 3. SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-8a887f12 (Free tier eligible, 64-bit).

Select a free tier eligible instance type, launch

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input checked="" 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 type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Low to Moderate

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

AMI Details [Edit AMI](#)

 Amazon Linux AMI 2017.09.0 (HVM), SSD Volume Type - ami-e689729e

Free tier eligible The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

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.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name	Description
launch-wizard-2	launch-wizard-2 created 2017-10-13T12:26:17.764-04:00

[Cancel](#) [Previous](#) [Launch](#)

Create key pair & launch

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
ISACA_workshop

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.

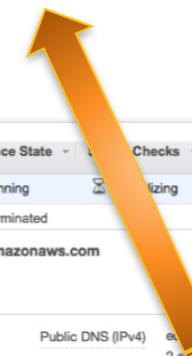
[Cancel](#)

[Launch Instances](#)



IPv4 Public IP 34.215.204.5

Instance ID	Instance Type	Availability Zone	Instance State	Checks	Alarm Status	Public DNS (IPv4)
66822a015e6f24	t2.micro	us-west-2b	running	-	None	ec2-34-215-204-5.us-west-2.compute.amazonaws.com
560f268cbcfbd2	t2.micro	us-west-2b	terminated	-	None	-
e6f24	Public DNS: ec2-34-215-204-5.us-west-2.compute.amazonaws.com					
	Details Monitoring Tags					
	Instance ID: i-04a66822a015e6f24					
	State: running					
	Type: t2.micro					
	Public IPs:					
	Zone: us-west-2b					
	Groups: launch-wizard-2, view inbound rules					
	Events: No scheduled events					
	AMI ID: amzn-ami-hvm-2017.09.0.20170930-x86_64-gp2 (ami-e689729e)					
	Platform: -					
	IAM role: -					
	Network interfaces: eth0					
	Source/dest. check: True					



~/.ssh/config shortcut

```
host isaca
  User ec2-user
  Hostname 34.215.204.5
  IdentityFile /Users/trice/Downloads/ISACA_workshop.pem
```

```
$ sudo yum -y update
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-
main
| 2.1 kB 00:00:00
amzn-
updates
| 2.5 kB 00:00:00
Resolving Dependencies
--> Running transaction check
---> Package amazon-ssm-agent.x86_64 0:2.1.4.0-1.amzn1 will be updated
```

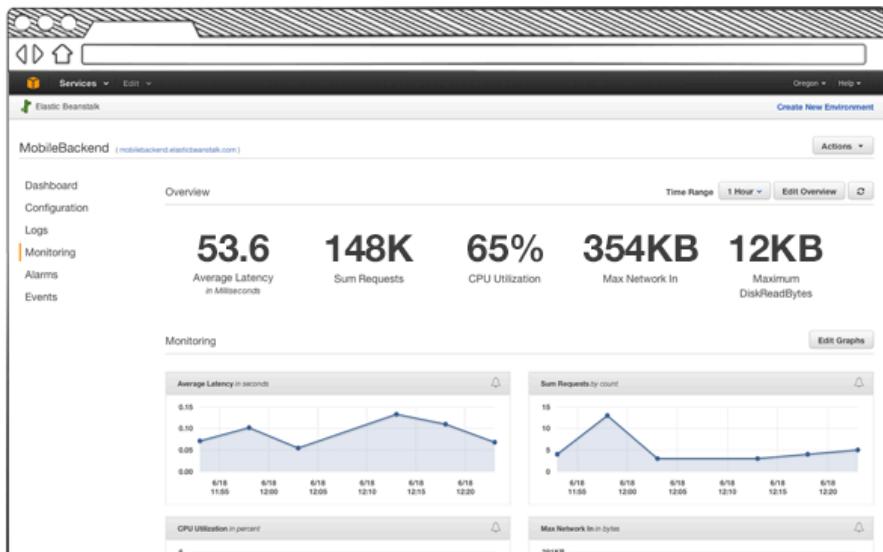
Login!

```
$ chmod 600 ~/Downloads/ISACA_workshop.pem
$ ssh -i ~/Downloads/ISACA_workshop.pem ec2-user@34.215.204.5
Warning: No xauth data; using fake authentication data for X11
forwarding.
X11 forwarding request failed on channel 0
```



```
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
No packages needed for security; 5 packages available
Run "sudo yum update" to apply all updates.
```

Elastic Beanstalk: Provision



Welcome to AWS Elastic Beanstalk

With Elastic Beanstalk, you can **deploy**, **monitor**, and **scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

To deploy your **existing web application**, create an [application source bundle](#) and then [create a new application](#). If you're using **Git** and would prefer to use it with our command line tool, please see [Getting Started with the CLI](#).

To deploy a **sample application** with just one click, select a platform and click **Launch Now**.

By launching the **sample application**, you allow AWS Elastic Beanstalk to administer AWS resources and necessary permissions on your behalf. [Learn more](#)

Python

Looking for a different platform? [Let us know](#).

AWS Elastic Beanstalk will provision an environment running Python 3.4 on 64bit Amazon Linux 2017.03 v2.5.2. Change the version.

Launch Now



Elastic Beanstalk: Provision

The screenshot shows the AWS IAM 'Create role' wizard. Step 1 (Trust) is active, showing a list of existing roles and their trusted entities. Step 2 (Permissions) and Step 3 (Review) are shown as part of the flow.

Create role

Search: Q Search

Role name	Description	Trusted entities
aws-elasticbeanstalk-ec...	AWS service: elasticbeanstalk.amazonaws.com	
aws-elasticbeanstalk-se...	AWS service: elasticbeanstalk.amazonaws.com	
rds-monitoring-role	AWS service: rds.amazonaws.com	
twitterapiusage	AWS service: elasticbeanstalk.amazonaws.com	

Create role

1 Trust **2 Permissions** **3 Review**

Role name* Jenkins Access
Maximum 64 characters. Use alphanumeric and '+,-,_,-' characters.

Role description
Maximum 1000 characters. Use alphanumeric and '+,-,_,-' characters.

Trusted entities AWS service: elasticbeanstalk.amazonaws.com

Policies AWSElasticBeanstalkService AWSElasticBeanstalkEnhancedHealth

* Required **Create role**

<https://aws.amazon.com/blogs/devops/building-continuous-deployment-on-aws-with-aws-codepipeline-jenkins-and-aws-elastic-beanstalk/>

Add user

1 Details 2 Permissions 3 Review 4 Complete

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* [Add another user](#)

Select AWS access type

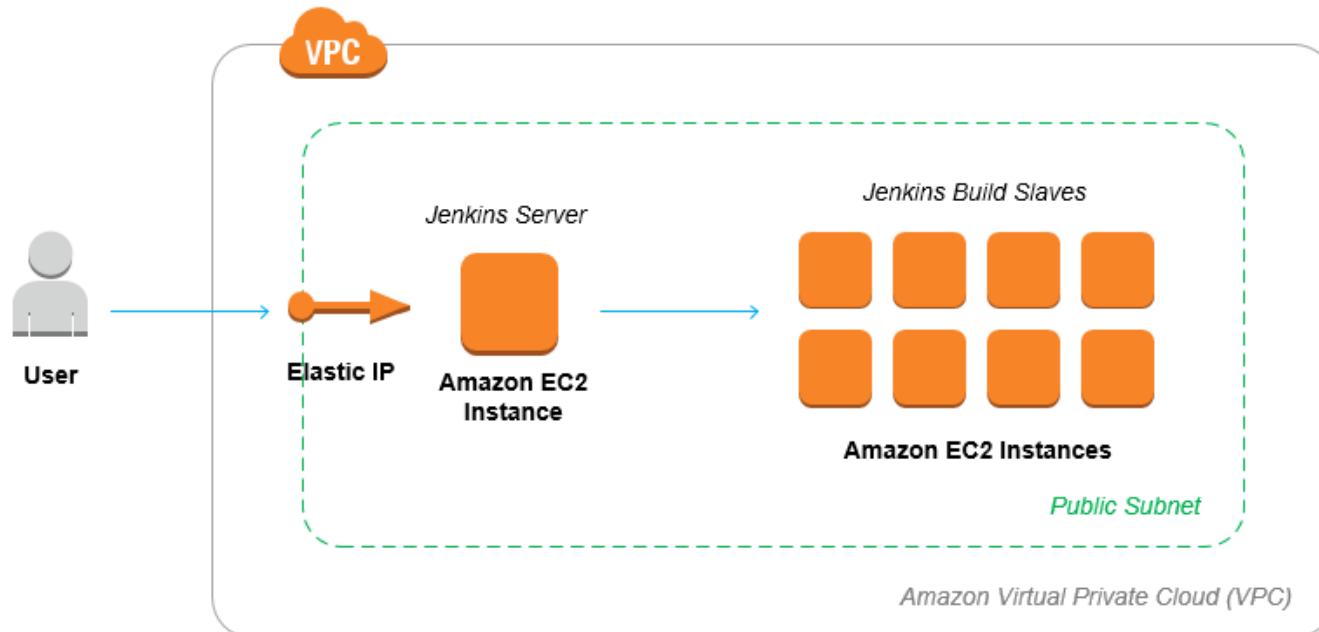
Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* Programmatic access
Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access
Enables a password that allows users to sign-in to the AWS Management Console.

Filter: Policy type ▾		<input type="text" value="ElasticB"/>
	Policy name ▾	
<input type="checkbox"/>	▶ AWSElasticBeanstalkCu...	
<input type="checkbox"/>	▶ AWSElasticBeanstalkEn...	
<input checked="" type="checkbox"/>	▶ AWSElasticBeanstalkFul...	
<input type="checkbox"/>	▶ AWSElasticBeanstalkM...	
<input type="checkbox"/>	▶ AWSElasticBeanstalkRe...	

Jenkins



Jenkins: installation

```
$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
http://pkg.jenkins-ci.org/redhat/jenkins.repo
```

```
$ sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key
```

```
$ sudo yum install git -y
```

```
$ sudo yum install jenkins -y
```

```
$ sudo service jenkins start
```

```
Starting Jenkins
```

```
$
```

[OK]

If you get Java8 errors

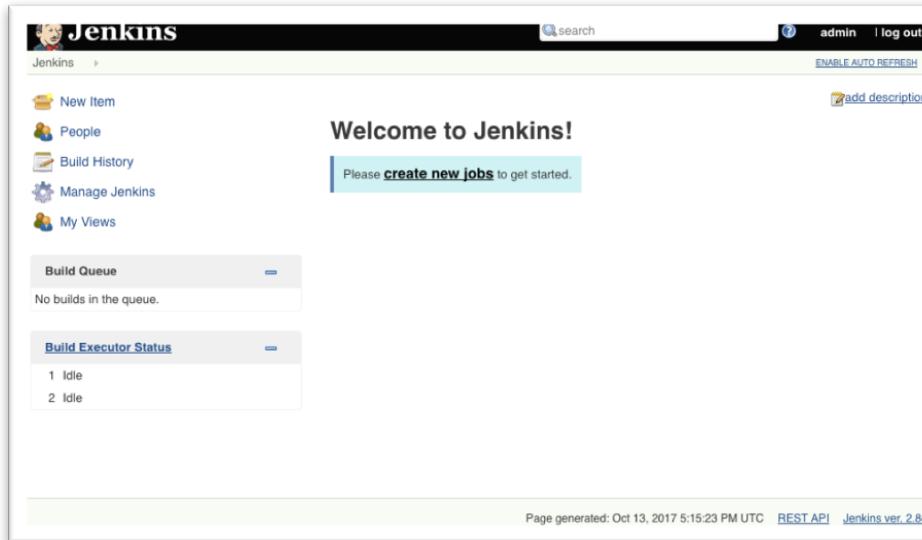
```
$ sudo service jenkins start
Starting Jenkins Jenkins requires Java8 or later, but you are running 1.7.0_151-
mockbuild_2017_08_09_21_42-b00 from /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.151.x86_64/jre
java.lang.UnsupportedClassVersionError: 51.0
at Main.main(Main.java:124)
$ sudo yum install -y java-1.8.0
$ sudo yum remove -y java-1.7.0-openjdk
$ sudo service jenkins start
Starting Jenkins
$
```

[OK]

Jenkins: Login

From EC2 Linux prompt

```
$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
somestring
```



Jenkins: Plugins

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links for New Item, People, Build History, Manage Jenkins, and My Views. Below that are two collapsed sections: Build Queue (No builds in the queue) and Build Executor Status (1 Idle). The main content area is titled 'Manage Jenkins' and lists several configuration items with icons:

- Configure System (gear icon)
- Configure Global Security (padlock icon)
- Global Tool Configuration (wrench and screwdriver icon)
- Reload Configuration from Disk (refresh icon)
- Manage Plugins (puzzle piece icon)
- System Information (monitor icon)
- System Log (clipboard icon)
- Load Statistics (ECG icon)

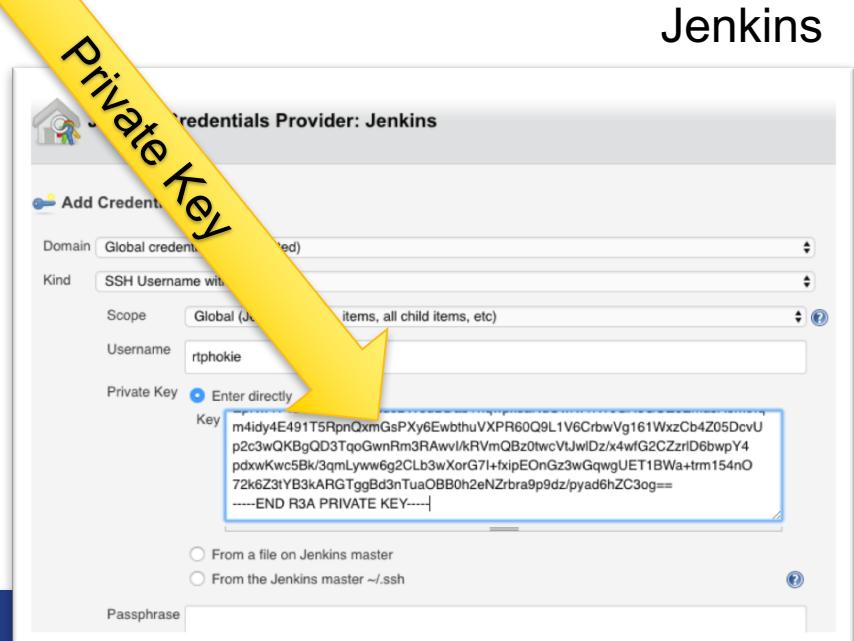
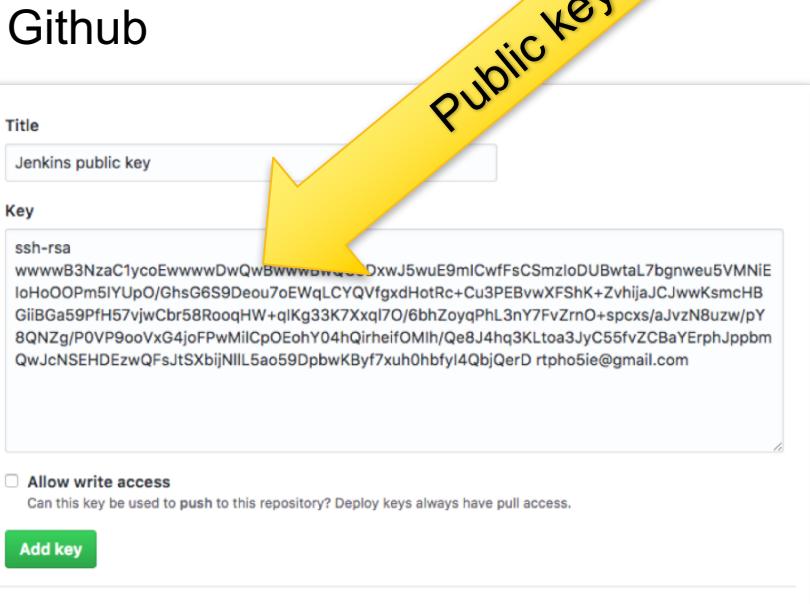
A large yellow arrow points to the 'Manage Plugins' link.

Install these plugins

- AWS Elastic Beanstalk Deployment Plugin (not Publisher)
- GitHub Authentication

Github: Deploy key

```
$ ssh-keygen -t rsa -C 'trice@cisco.com'
```



Apply Public Key to Github

Title
Jenkins public key

Key
ssh-rsa
wwwB3NzaC1ycoEwwwwDwQwBwwwBwQC9DxwJ5wuE9mlCwfFsCSmzloDUBwtal7bgnweu5VMNiE
IoHoOOPm5IYUpO/GhsG6S9Deou7oEwqLCYQVfgxdHotRc+Cu3PEBvwXFShK+ZvhijaJCJwwKsmcHB
GiIBGa59PfH57vjqCbr58RooqHW+qlKg33K7XxqJ70/6bhZoyqPhL3nY7FvZrnO+spcxsaJvzN8uzw/pY
8QNZg/P0VP9ooVxG4joFPwMilCpOEohY04hQirheifOMlh/Qe8J4hq3KLtoa3JyC55fvZCBaYErphJppbm
QwJcNSEHDEzwQFsJtSXbijNILL5ao59DpbwKByf7xuh0hblyl4QbJQerD rpho5ie@gmail.com

Allow write access
Can this key be used to push to this repository? Deploy keys always have pull access.

Add key

- Settings >> Deploy keys
- Click Add Deploy Key
enter public key
enter a title

Apply Private Key to Jenkins Job

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: SSH Username with private key

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: rtphokie

Private Key: Enter directly
Key:
m4idy4E491T5RpnQxmGsPXy6EwbthuVXPR60Q9L1V6CrbwVg161WxzCb4Z05DcvUp2c3wQKBgQD3TqoGwnRm3RAwv1kRVmQBz0twcVtJwlDz/x4wfG2CZzrlD6bwPY4pdxwKwc5Bk/3qmLyww6g2CLb3wXorG7l+fxipEOnGz3wGqwgUET1BWa+trm154nO72k6Z3tYB3kARGTggBd3nTuaOBBoh2eNZrbra9p9dz/pyad6hZC3og==
----END RSA PRIVATE KEY----

From a file on Jenkins master
 From the Jenkins master ~/.ssh

Passphrase:

- New Freestyle Job
- Source Code Mgmt->Git url:
<https://github.com/rtpohkie/hello-world.git>
- Click add key
SSH with key
enter userid
enter Private key
click add

Jenkins: deploy to EB post build

AWS Elastic Beanstalk

AWS Credentials and Region

Credentials

AWS Region

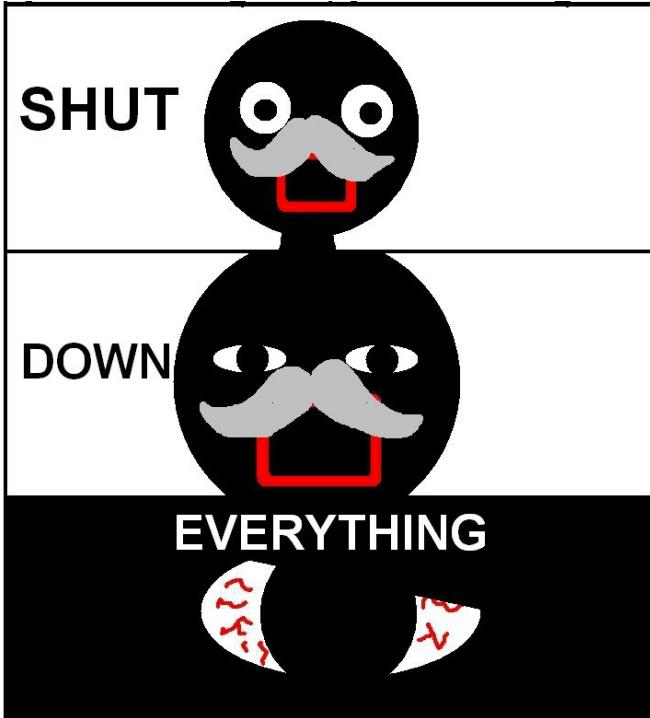
Number Of Attempts

Application and Environment

Application Name

Environment Name(s)

Before you go



AWS bills based on running and stopped instances. To avoid unexpected bills:

1. Shutdown your EC2 instances and Elastic Beanstalk apps.
2. Terminate them ASAP

Thank you.

trice@cisco.com

