



# 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](http://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



SEC-USR-MESS-2 SEC-RUN-SAFEC SEC-RUN-OSC SEC-RUN-ASLR SEC-PWD-MAXLIFE SEC-PWD-AUDIT SEC-OPS-RDY SEC-OPS-NPROD SEC-NRCV-CRED-5 SEC-LOG-TENISOL-2 SEC-LOG-STATCHG SEC-LOG-LOGCONF SEC-LOG-LIMIT SEC-LOG-CONTENT SEC-LOG-ATTACK SEC-LOG-ADMIN SEC-IPS-ESP-5 SEC-INSEC-USER SEC-CSP-LOGTRUST SEC-CRY-LOG SEC-AUT-LOGZCHG SEC-AUT-LOGNCHG SEC-WEB-HTTPMETH-2 SEC-SRT-311000-3 SEC-OPS-BUSCONT-2 SEC-LOG-CENTRAL-3 SEC-INF-HARDEN-2 SEC-509-FQDN-2 SEC-SET-TIME SEC-SRT-311000-2 SEC-LOG-INDC-2 SEC-SRT-DTLS-2 SEC-CHG-LOGD-3 SEC-CRE-NOLOCK SEC-IP-IPv6 SEC-AUD-FIELD-3 SEC-OPS-BUSCONT SEC-INF-DOS SEC-INF-AVSCAN SEC-OPS-REVOKE SEC-DAT-RETAIN SEC-DAT-REMOVE SEC-DAT-AUDIT SEC-WEB-NOREDIR SEC-WEB-IDCACHE SEC-WEB-IDCTX SEC-SDP-311030 SEC-SRT-311020 SEC-SRT-311010 SEC-TIME-NTP SEC-ASU-TRAIN-2 SEC-WEB-SMTIME-2 SEC-WEB-STATE SEC-WEB-CRYPTOCOOKIE-2 SEC-WEB-RESP-2 SEC-WEB-SESCOOKIE-2 SEC-OPS-MAXLIFE SEC-DAT-VMCRYPT SEC-WEB-IDATTR-2 SEC-AUT-CREROT SEC-LOG-CENTRAL-2 SEC-AUT-LOG SEC-LOG-RET SEC-WEB-NPCOOKIE-2 SEC-DAT-SANITIZE SEC-DAT-KEYMGMT SEC-ALL-LOGD-2 SEC-SW-NEEDONLY SEC-ASU-STATIC SEC-HTTP-SSL3-3 SEC-ASU-TMOD SEC-OPS-LEGAL SEC-OPS-INCOPOL SEC-INF-MGTZONE SEC-CRE-PRI V SEC-CRE-MULTIFAC SEC-DAT-GEOG SEC-DAT-SEGR SEC-DAT-BACKUP SEC-DAT-MEDIA SEC-WEB-CLKJACK SEC-WEB-IDVALID SEC-PRV-KNOWWHAT SEC-PRV-NOCOLL SEC-PRV-USERAUTH SEC-PRV-ERASE SEC-DAT-PRIV-2 SEC-PRV-LOCALID SEC-PRV-MANAGE SEC-TLS-CURR-3 SEC-LOG-FORM SEC-OPS-RESTCRYPT-2 SEC-WEB-XSSUBCXT-2 SEC-WEB-HTTPMETH SEC-WEB-SQLIN-2 SEC-509-CHAIN SEC-DAT-BACKUPTS SEC-LOG-TRANS SEC-DAT-SOP SEC-WEB-URLPARAM SEC-WEB-SYSIN-2 SEC-WEB-ACCBYURL-2 SEC-LOG-PROTO SEC-LOG-APPREQ SEC-WEB-IDALL-2 SEC-LOG-ACCESS-2 SEC-WEB-CSRF SEC-DAT-BACKUPER SEC-DAT-BACKUPI SEC-509-LIFETIME SEC-509-CERTEXT SEC-CR E-LIMTRY-2 SEC-LOG-NOSENS-2 SEC-WEB-INJIN-2 SEC-WEB-SANIHTML SEC-WEB-AUTOCOMP SEC-DAT-BACKUPET SEC-LOG-INFRAREQ SEC-WEB-ID-2 SEC-DAT-ISMS-2 SEC-WEB-XSS SEC-509-FQDN SEC-HTTP-HSTS SEC-DAT-MONEY SEC-WEB-XPATHIN-2 SEC-509-REVOKE SEC-SUP-PATCH SEC-UPS-REGI SEC-INT-CRED-2 SEC-DEF-CRED-2 SEC-OPS-CHGMGT SEC-WEB-SCAN SEC-INF-SCAN SEC-INF-THIRD SEC-INF-DEVNET SEC-INF-ZONES SEC-INF-MONITOR SEC-INF-FWSEP SEC-INF-VER SEC-INF-HARDEN SEC-OPS-ASTMGT-2 SEC-OUT-CRED-2 SEC-OPS-STRENGTH SEC-CRY-PRIM SEC-OPS-SEG-2 SEC-AUT-ACCDEF SEC-CRY-STDCODE SEC-CRE-SHARE SEC-OPS-SUDO SEC-AUT-AUTH SEC-OPS-PUBCRYPT-2





Triangle  
InfoSeCon

[triangleinfosecon.com](http://triangleinfosecon.com)



Friday October 27, 2017  
Raleigh Convention Center, Raleigh, NC

# In this workshop we will setup

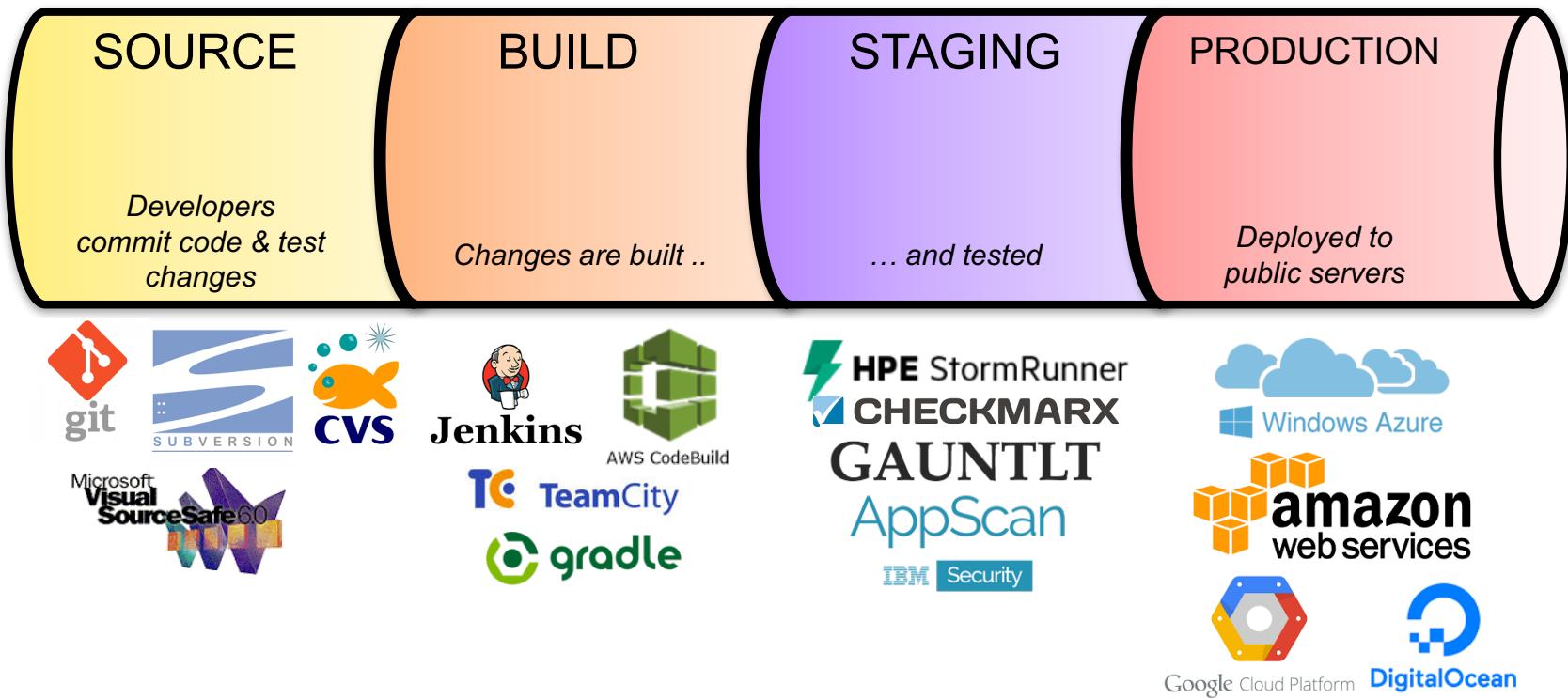
- A source code repository
- A centralized build server
- Deploy a simple micro service into the cloud



Along the way, we'll form them into a continuous integration toolchain ... securely



# Code pipeline



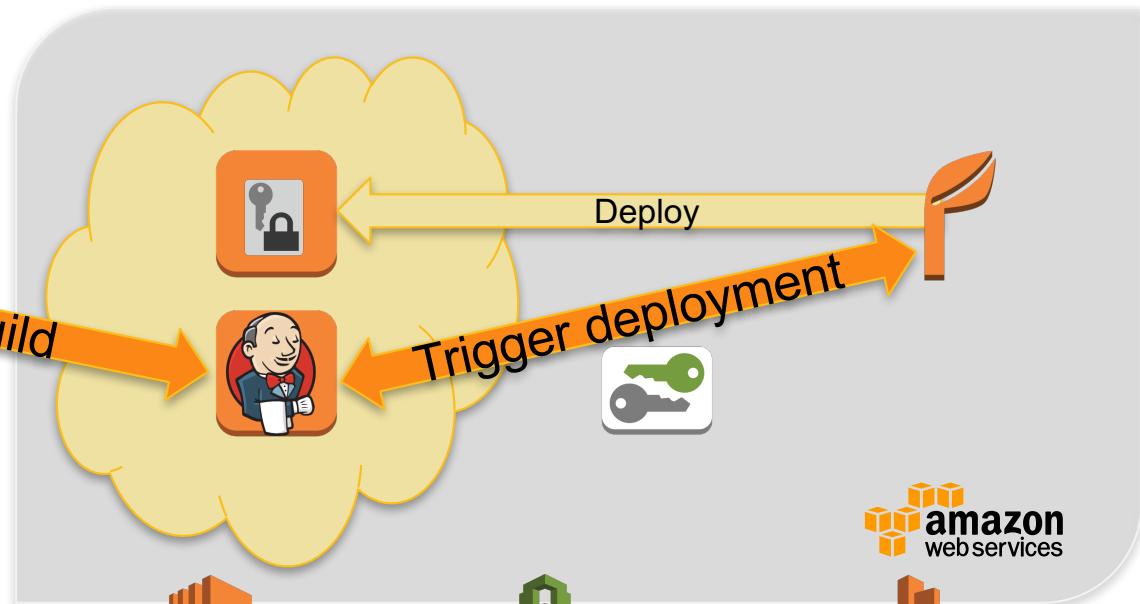
# Our Continuous Delivery Pipeline

Source Code Management



GitHub

Build      Identity      Deployment



# 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

# Get the app source code from GitHub

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

Pull ...  
then clone

Clone

The screenshot shows the GitHub repository page for `rtpphokie/hello-world`. At the top, there's a navigation bar with links for This repository, Search, Pull requests, Issues, Marketplace, and Explore. Below the navigation is a header for the repository, showing it has 1 star, 0 forks, and 0 issues/pull requests/projects/wiki/insights/settings. A message says "No description, website, or topics provided." and there's a link to "Edit". Below this is a summary bar with metrics: 4 commits, 1 branch, 0 releases, 1 contributor, and a license of GPL-3.0. A "New pull request" button is also present. The main area displays a list of commits:

File	Description	Time
<code>README</code>	updated README	Latest commit de7baf0 4 minutes ago
<code>.gitignore</code>	Initial commit	2 hours ago
<code>LICENSE</code>	Initial commit	2 hours ago
<code>README.md</code>	updated README	4 minutes ago
<code>application.py</code>	converted to a microservice doing something useful	5 minutes ago
<code>ncaa.dat</code>	converted to a microservice doing something useful	5 minutes ago

# 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

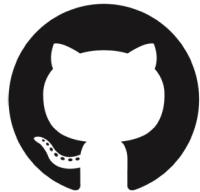
This is a RESTful micro service returning information on NCAA Div I football teams

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

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

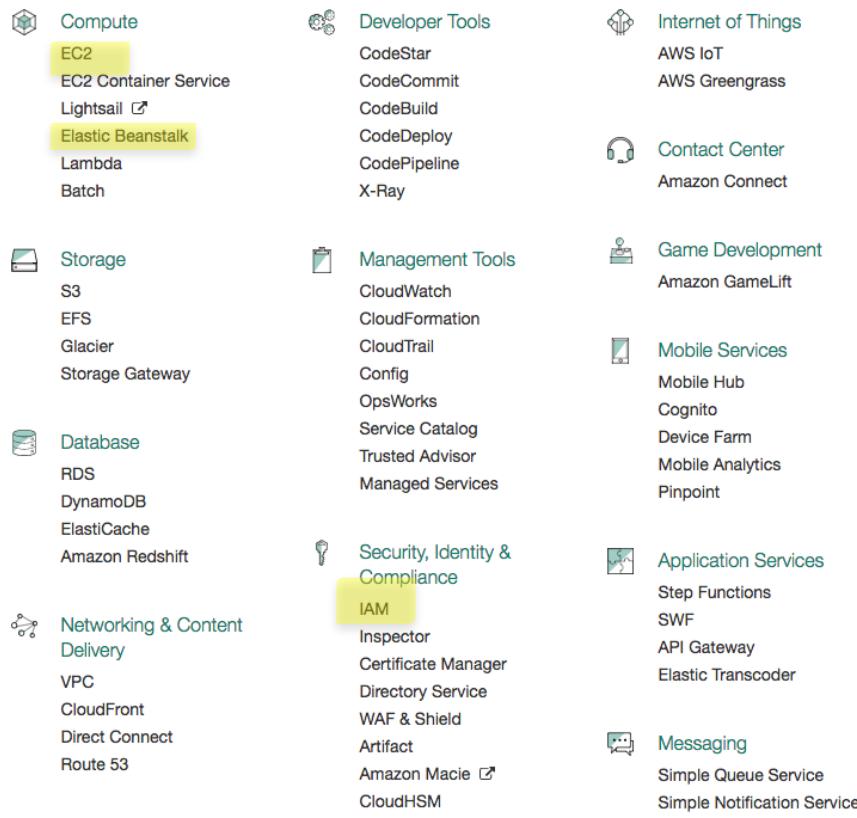
# Our Continuous Delivery Pipeline

Source Code  
Management



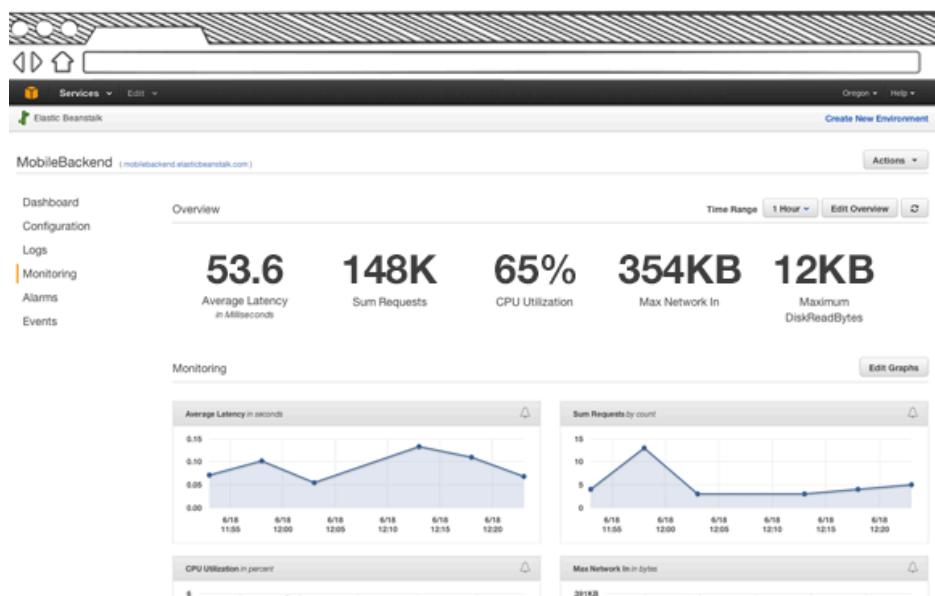
GitHub

# AWS console



- We will be using
- EC2
    - Security Groups
    - Instances
  - IAM
  - Elastic Beanstalk

# 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 EB 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

2.7 on 64 Bit Linux

Working for a different platform? Let us know.

AWS Elastic Beanstalk will create a new application environment running Python 3.4 on 64bit Amazon Linux 2017.03 v2.5.2. [Change platform version](#).

**Launch Now**

# Elastic Beanstalk provision

- Create Environment
  - Create Webserver (Python, change to 2.7; single instance)
  - Use sample application
  - Launch

Environment Information

Enter your environment information.

Environment name:

Environment URL:

Description:  Optional: 200 character maximum

# IAM: Provision EB User

AWSElasticBeanstalkFullAccess

Add user Delete user

User name: Jenkins2EB

Select AWS access type

Access type\*:  Programmatic access  AWS Management Console access

Set permissions for Jenkins2EB

Add user to group Copy permissions from existing user Attach existing policies directly

Create policy Refresh

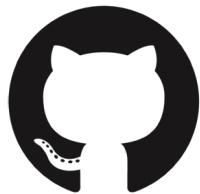
Policy name	Type	Attachments	Description
AWSElasticBeanstalkFullAccess	AWS managed		

Previous Next: Review

Cancel Previous Create user

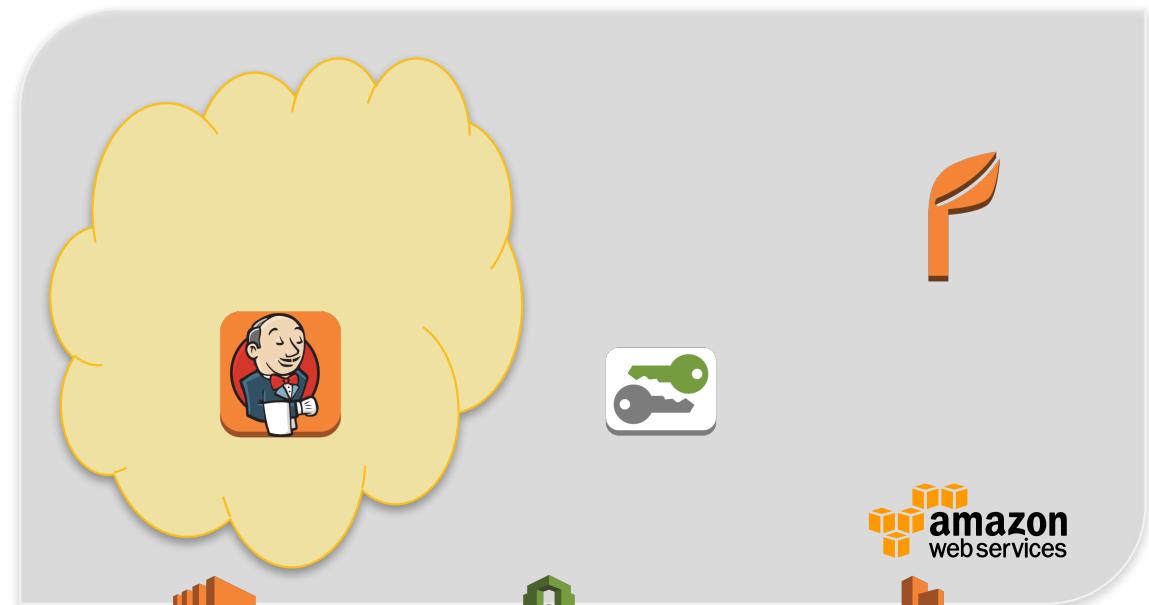
# Our Continuous Delivery Pipeline

Source Code Management



GitHub

Build



Identity

Deployment

# EC2: 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				

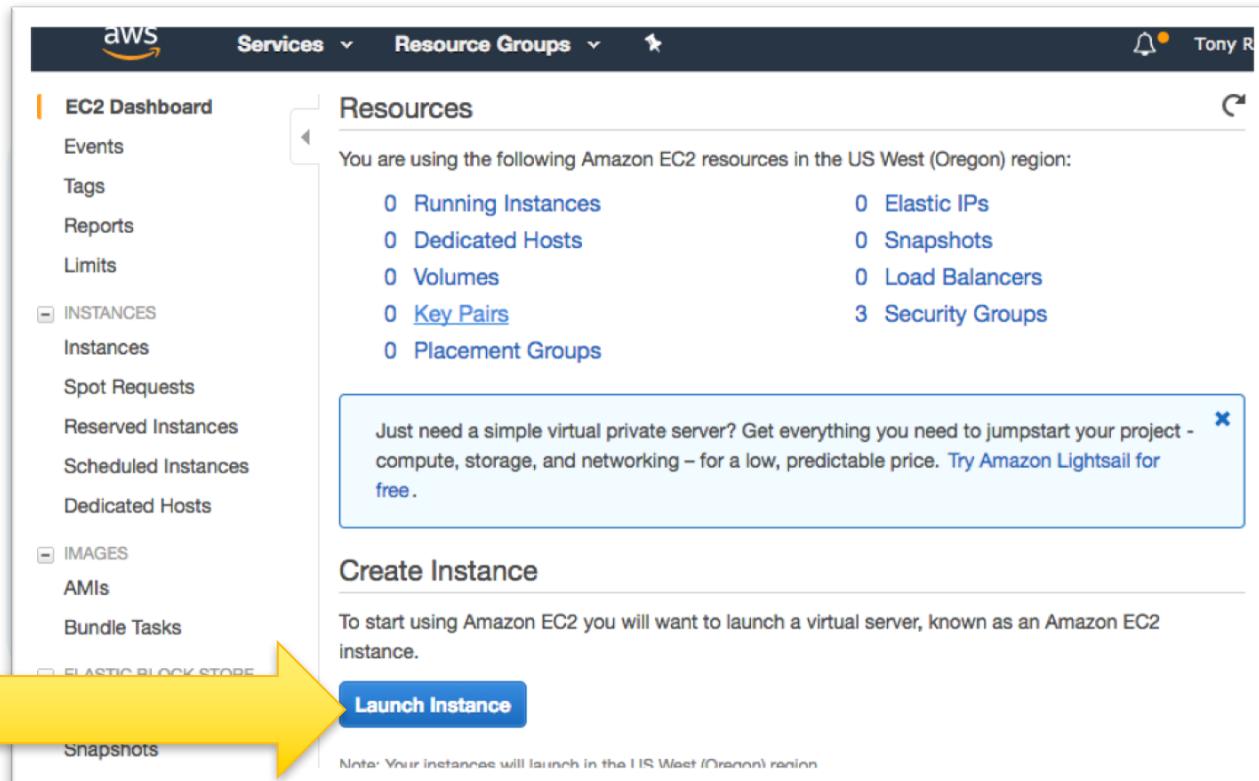
Add Rule

Cancel Create



Add access to port 22  
While you are creating this

# EC2: launch EC2 instance



# AWS: launch EC2 instance

The image displays two screenshots of the AWS EC2 console. The left screenshot shows the EC2 Dashboard with a yellow arrow pointing to the 'Launch Instance' button. The right screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' screen, with a yellow arrow pointing to the 'Select' button next to the Amazon Linux AMI entry.

**EC2 Dashboard (Left Screenshot):**

- Resources
- 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
  - 3 Security Groups
- 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.
- Create Instance
- To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
- Launch Instance**

**Step 1: Choose an Amazon Machine Image (AMI) (Right Screenshot):**

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group

Step 1: Choose an Amazon Machine Image (AMI)

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.

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs

Free tier only i

Image	Name	Type	Select	64-bit
	Amazon Linux AMI 2017.09.0 (HVM), SSD Volume Type - ami-e689729e	Free tier eligible	<b>Select</b>	64-bit
	Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-9fa343e7	Free tier eligible	<b>Select</b>	64-bit
	SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-8a887ff2	Free tier eligible	<b>Select</b>	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 Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	

**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 run 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 security group:  Create a new security group  Select an existing security group

Security Group	Name	Description
sg-fd7aed80 awseb-e-gpsjv2ymgz-stack-AWSEBLoadBalancerSecurityGroup-1LYQMCTI3FX3UE	AWS Elastic Load Balancer Security Group	Elastic Beanstalk created security group used when no ELB security group is specified.
sg-f070e78d awseb-e-gpsjv2ymgz-stack-AWSEBSecurityGroup-1MDV	AWS Elastic Beanstalk Security Group	Security group for Elastic Beanstalk environment.
sg-e46d7586default	default VPC security group	
sg-ce74e3b3Jenkins8080	Jenkins8080	

Inbound rules for sg-ce74e3b3 (Selected security groups: sg-ce74e3b3)

Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	8080	0.0.0.0/0	Jenkins port 8080
Custom TCP Rule	TCP	8080	::/0	Jenkins port 8080

**Review and Launch**

**Review and Launch**

# Create key pair & launch

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

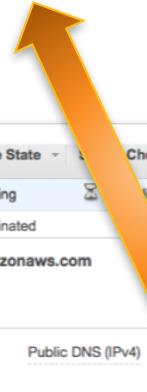
A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key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

ISACA\_workshop

**Info:** 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.

IPv4 Public IP **34.215.204.5**



Instance ID	Instance Type	Availability Zone	Instance State	Check	Alarm Status	Public DNS (IPv4)
66822a015e6f24	t2.micro	us-west-2b	running	Passing	None	ec2-34-215-204-5.us-west-2.compute.amazonaws.com
560f268cbcfbd2	t2.micro	us-west-2b	terminated	Passing	None	
e6f24	Public DNS: ec2-34-215-204-5.us-west-2.compute.amazonaws.com					
	Checks	Monitoring	Tags			
Instance ID	i-04a66822a015e6f24			Public DNS (IPv4)	ec2-34-215-204-5.us-west-2.compute.amazonaws.com	
InstanceState	running			IPv4 Public IP	34.215.204.5	
Instance Type	t2.micro			IPv6 IPs	-	
Private IPs				Private DNS	ip-172-31-33-155.us-west-2.compute.internal	
Zone	us-west-2b			Private IPs	172.31.33.155	
Groups	launch-wizard-2, view inbound rules			Secondary private IPs		
Events	No scheduled events			VPC ID	vpc-4ac6d728	
AMI ID	amzn-ami-hvm-2017.09.0.20170930-x86_64-gp2 (ami-e689729e)			Subnet ID	subnet-ba0315d8	
Platform	-			Network interfaces	eth0	
IAM role	-			Source/dest. check	True	

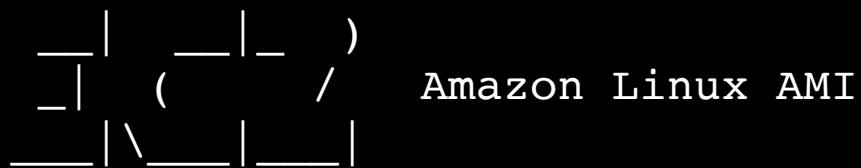
## **~/.ssh/config shortcut**

```
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
```

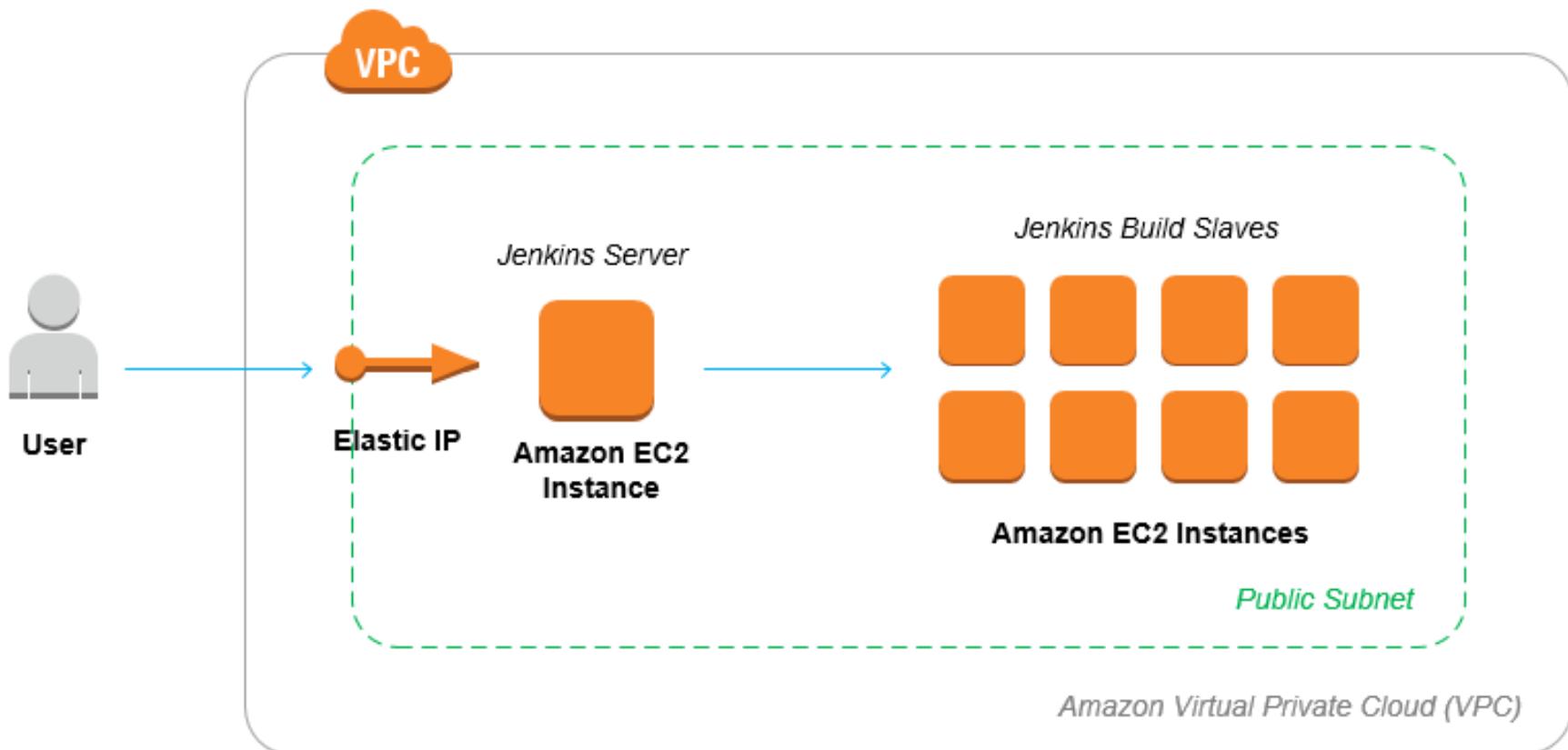
# Login!

```
$ 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.
```

# Jenkins



# Jenkins: installation

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

```
$ 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 -y jenkins
```

```
$ sudo service jenkins start  
Starting Jenkins
```

[ OK ]

# If you get Java8 errors

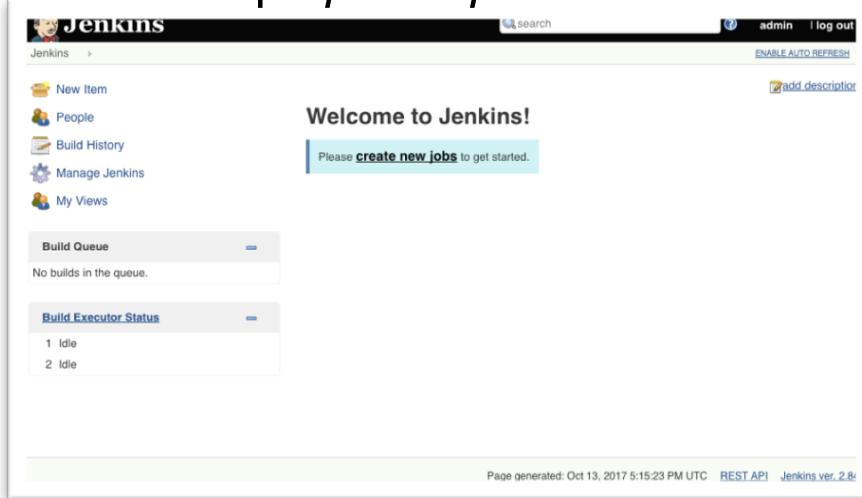
```
$ sudo service jenkins start
Starting Jenkins Jenkins requires Java8 or later, but you are running 1.7.0_151-
mockbuild_2017_08_09_21_42-b00 from /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.151.x86_64/jre
java.lang.UnsupportedClassVersionError: 51.0
at Main.main(Main.java:124)
$ sudo yum -y install java-1.8.0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

<http://publicip:8080>



# Jenkins: Plugins

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links: New Item, People, Build History, Manage Jenkins (which is selected), and My Views. Below these are two collapsed sections: Build Queue (No builds in the queue) and Build Executor Status (1 Idle, 2 Idle). The main content area is titled "Manage Jenkins" and contains several configuration links: Configure System, Configure Global Security, Global Tool Configuration, Reload Configuration from Disk, Manage Plugins (with a yellow arrow pointing to it), System Information, System Log, and Load Statistics.

## Install these plugins

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

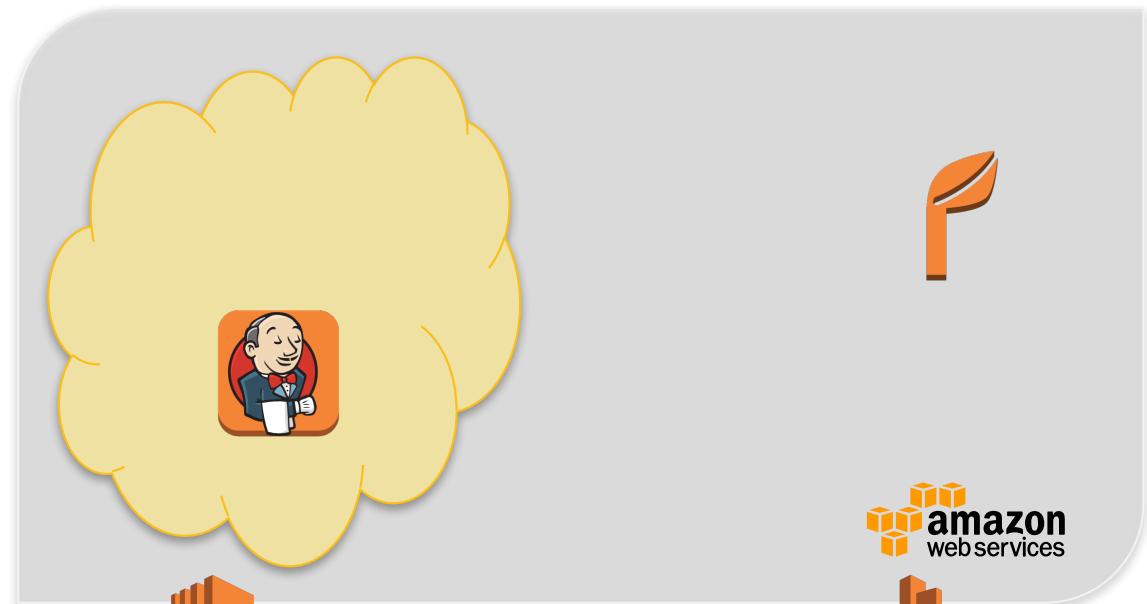
# Our Continuous Delivery Pipeline

Source Code  
Management



GitHub

Build



Identity

Deployment

## Create a key pair

- Login to EC2 instance
- sudo su – jenkins
- ssh-keygen -t rsa
- Public and private is\_rsa files are in ~jenkins/.ssh

# Create a build job

The screenshot shows the Jenkins 'Create a build job' interface. On the left, there's a sidebar with a 'Freestyle project' section and a main panel for 'Source Code Management'. In 'Source Code Management', 'Git' is selected, and a 'Repository URL' field contains 'http://rtp...'. On the right, a modal window titled 'Build' is open for 'AWS Elastic Beanstalk'. It contains sections for 'AWS Credentials and Region' (with credentials 'AKIAJBOP35EQ67KDT2AA' and region 'us-west-2'), and 'Application and Environment' (with application name 'My First Elastic Beanstalk Application' and environment 'Default-Environment').

Enter an item name  
HelloNCAA  
Required field

Freestyle project  
This is the central feature of Jenkins. You can use it to build almost anything.

Source Code Management

None  
Git

Repositories

Repository URL http://rtp...

Credentials

Build

AWS Elastic Beanstalk

AWS Credentials and Region

Credentials AKIAJBOP35EQ67KDT2AA (Jenkins to Elastic Bean...) Add

AWS Region us-west-2

Number Of Attempts 30

Validate Credentials

Application and Environment

Application Name My First Elastic Beanstalk Application

Environment Name(s) Default-Environment

Validate Coordinates

# Github: Deploy key

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

Github

Public key

Title  
Jenkins public key

Key

```
ssh-rsa
wwwB3NzaC1yc0EwwwDwQwBwwwDwQwC...DxwJ5wuE9mlCwfFsCSmzloDUBwtaL7bgnweu5VMNiE
loHoOOPm5lYUpO/GhsG6S9Deou7oEWqLCYQVfgxdHotRc+Cu3PEBvwXFShK+ZvhijaJCJwwKsmcHB
GiiBga59PfH57vJwCbr58RooqHW+qlKg33K7Xxqj7O/6bhZoyqPhL3nY7FvZrnO+spcxas/aJvzN8uzw/pY
8QNzg/P0VP9ooVxG4joFPwM1CpOEohY04hQirheifOMlh/Qe8J4hq3KLtoa3JyC55fvZCBaYEphJppbm
QwJcNSEHDEzwQFsJtSXbijNILL5ao59DpbwKByf7xuh0hbfyI4QbjQerD rtpho5ie@gmail.com
```

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

**Add key**

Jenkins

Private Key

Add Credential

Domain Global (Job, workspace, items, all child items, etc)

Kind SSH Username with Private Key

Scope Global (Job, workspace, items, all child items, etc)

Username rtpphokie

Private Key  Enter directly

Key

```
m4idy4E491T5RpRpnQxmGsPxY6EwbthuVXPR60Q9L1V6CrbwVg161WxzCb4Z05DcvU
p2c3wQKBgQD3TqoGwnRm3RAwv/kRvmQBz0twcVtJwlDz/x4wfG2CZrlD6bwpY4
pdwxKwc5Bk/3qmLyww6g2CLb3wXorG7I+fxipEOnGz3wGqwgUET1BWa+trm154nO
72k6Z3tYB3kARGTggBd3nTuaOBBoh2eNrbr9p9dz/pyad6hZC3og==
-----END RSA PRIVATE KEY-----
```

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

Passphrase

# Apply Public Key to Github

Title  
Jenkins public key

Key  
ssh-rsa  
wwwB3NzaC1ycoEwwwwDwQwBwwwBwQC9DxwJ5wuE9mlCwfFsCSmzloDUBwtal7bgnweu5VMNiE  
loHoOPm5lYUpO/GhsGGS9Deou7oEWqLCYQVfgxdHotRc+Cu3PEBvxXFShK+ZvhijAJCJwwKsmcHB  
GiiBGa59Pfh57vjqwCbr58RooqHW+qIKg33K7Xxqj7O/6bhZoyqPhL3nY7FvZrnO+spcxas/aJvzN8uzw/pY  
8QNZg/P0VP9ooVxG4joFPwMiiCpOEohY04hQirheifOMlh/Qe8J4hq3KLtoa3JyC55fvZCBaYErphJppbm  
QwJcNSEHDEzwQFsJtSXbijNIIIL5ao59DpbwKByf7xuh0hbfyl4QbjQerD rtpho5ie@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

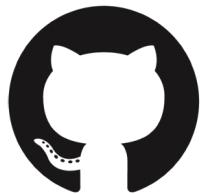
# Apply Private Key to Jenkins Job

The screenshot shows the Jenkins Credentials Provider interface. The title bar says "Jenkins Credentials Provider: Jenkins". Under the "Add Credentials" section, the "Kind" is set to "SSH Username with private key". The "Username" field contains "rphokie". The "Private Key" field has the radio button "Enter directly" selected, and the key value is pasted into the text area:  
m4idy4E491T5RpnQxmGsPxy6EwbthuVXPR60Q9L1V6CrbwVg161WxzCb4Z05DcvUp2c3wQKBgQD3TqoGwnRm3RAwvI/kRVmQBz0twcVtJwlDz/x4wfG2CZrlD6bwpY4pdxwKwc5Bk/3qmLyww6g2CLb3wXorG7l+fxipEOnGz3wGqwgUET1BWa+trm154nO72k623tYB3kARGTgg8d3nTuaOBBoh2eNzbra9p9dz/pyad6hZC3og==  
-----END RSA PRIVATE KEY-----  
Below the key field, there are two radio buttons: "From a file on Jenkins master" and "From the Jenkins master ~/.ssh". A "Passphrase" field is also present at the bottom.

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

# Our Continuous Delivery Pipeline

Source Code Management

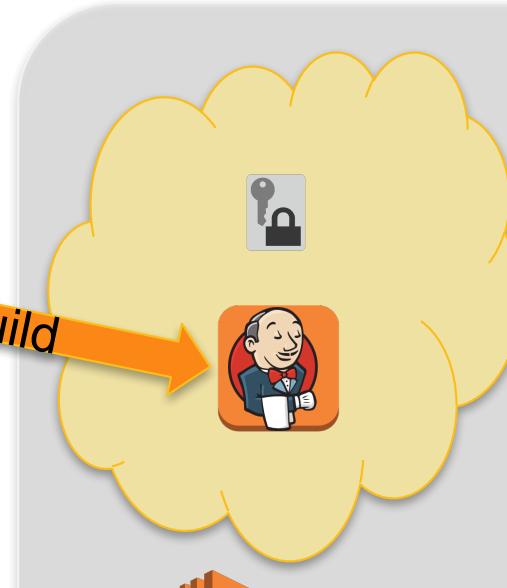


GitHub

*Trigger build*



Build



EC2

Identity



Deployment



Elastic  
Beanstalk

# Jenkins: deploy to EB post build

AWS Elastic Beanstalk

**AWS Credentials and Region**

- Credentials: AKIAJYHV6HXF567KPAD/ (highlighted with a yellow arrow)
- AWS Region: us-west-2
- Number Of Attempts: 30

**Application and Environment**

- Application Name: helloncaa (highlighted with a yellow arrow)
- Environment Name(s): helloncaa-env

**Version and Deployment**

- Version Label Format: \${BUILD\_TAG}

Select AWS Credentials

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: AWS Credentials

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

ID:

Description:

Access Key ID:

Secret Access Key:

IAM Role Support

Cisco Public 2017 All Rights Reserved

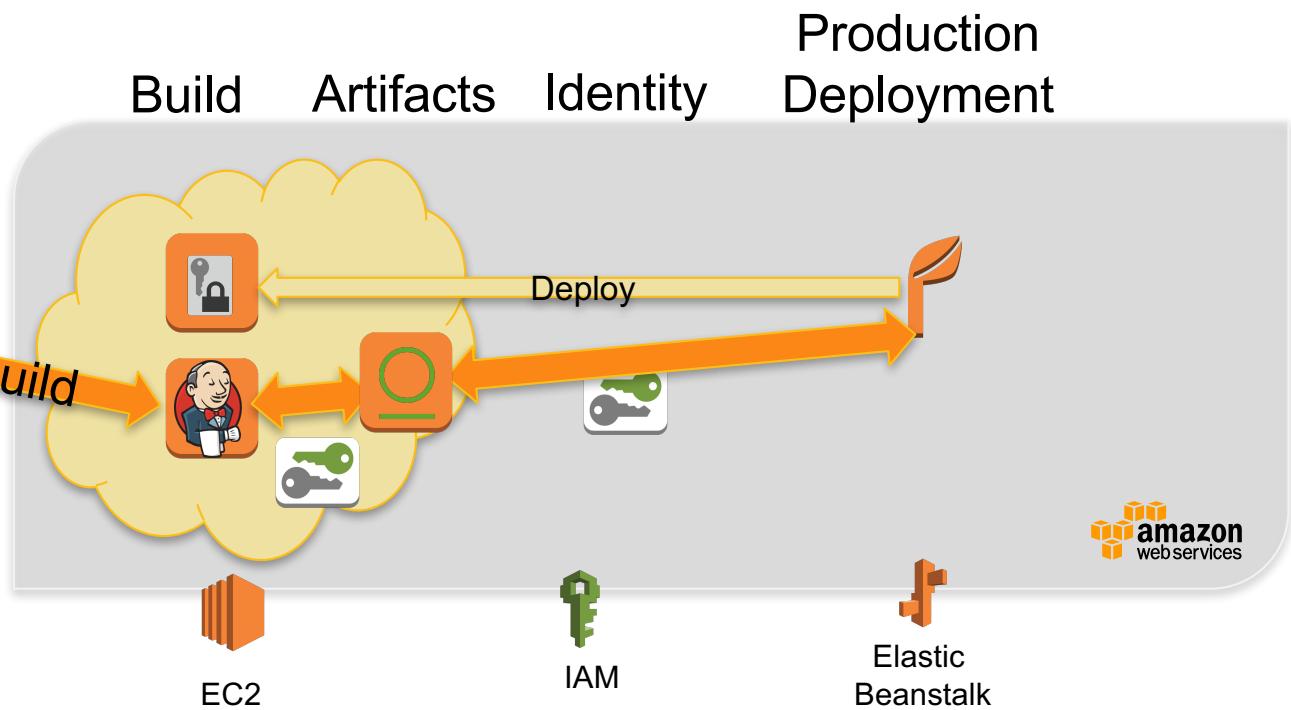
CISCO.

# Artifact Management

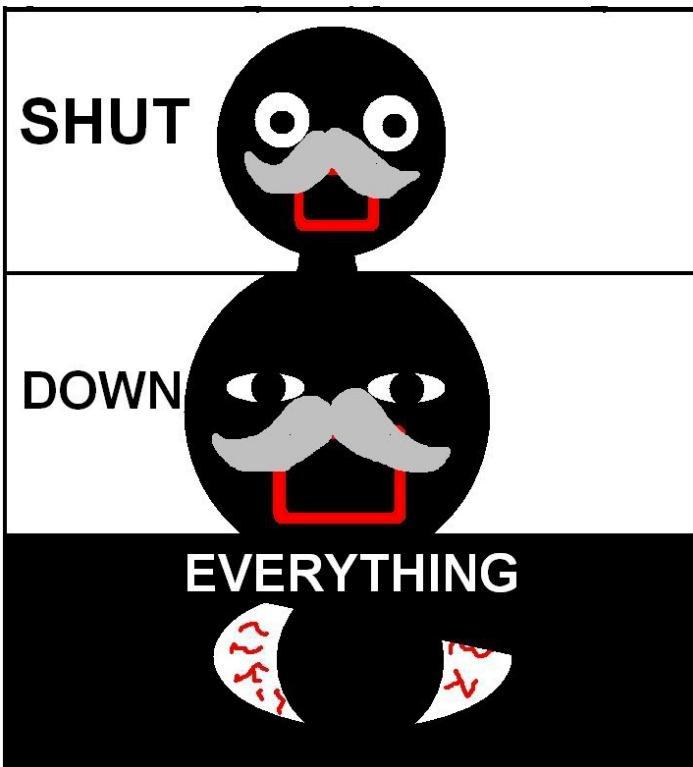
Source Code  
Management



GitHub



## 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](mailto:trice@cisco.com)

 @rtphokie

