Securing your DevOps Toolchain in the Cloud

Tony Rice

Cisco Security & Trust Organization

Prerequisites

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

Amazon AWS account (free tier is fine)

aws.amazon.com/free

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 Us

Tony Rice

Senior Information Security Engineer

RTP

Khadija Amin

Information Security Architect

SJC

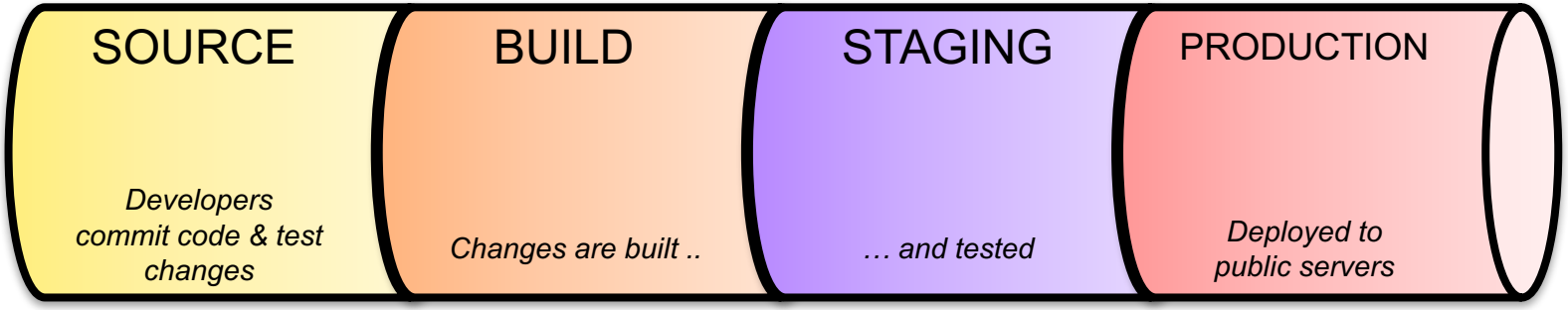
SEC-USR-MESS-2 SEC-RUN-SAFEC SEC-RUN-OSC SEC-RUN-ASLR SEC-PWD-MAXLIFE SEC-PWD-AUDIT SEC-OPS-RDY
SEC-OPS-NPROD SEC-NRCV-CRED-5 SEC-LOG-TENISOL-2 SEC-LOG-STATCHG SEC-LOG-LOGCONF SEC-LOG-LIMIT
SEC-LOG-CONTENT SEC-LOG-ATTACK SEC-LOG-ADMIN SEC-IPS-ESP-5 SEC-INSEC-USER SEC-CSP-LOGTRUST SEC-
CRY-LOG SEC-AUT-LOGZCHG SEC-AUT-LOGNCHG SEC-WEB-HTTPMETH-2 SEC-SRT-311000-3 SEC-OPS-BUSCONT-2 S
EC-LOG-CENTRAL-3 SEC-INF-HARDEN-2 SEC-509-FQDN-2 SEC-SET-TIME SEC-SRT-311000-2 SEC-LOG-INDC-2 S
EC-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-CRYPCOOK-2 SEC-WEB-RESP-2 SEC-WEB-SESCOOK-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-HTP-SSL3-3 SEC-ASU-TMOD SEC-OPS-LEGAL SEC-OPS-INCPOL SEC-INF-MGTZONE SEC-CRE-PRI
V SEC-CRE-MULTIFAC SEC-DAT-GEOG SEC-DAT-SEGR SEC-DAT-BACKUP SEC-DAT-MEDIA SEC-WEB-CLCKJACK SEC-
WEB-IDVALID SEC-PRV-KNOWWHAT SEC-PRV-NOCOLL SEC-PRV-USERAUTH SEC-PRV-ERASE SEC-DAT-PRIV-2 SEC-P
RV-LOCALID SEC-PRV-MANAGE SEC-TLS-CURR-3 SEC-LOG-FORM SEC-OPS-RESTCRYP-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-
URLPARM 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-HTP-HSTS SEC-DAT-MONE
Y 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-MONITO
R 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-PUBCRYP-2

In this workshop we will setup

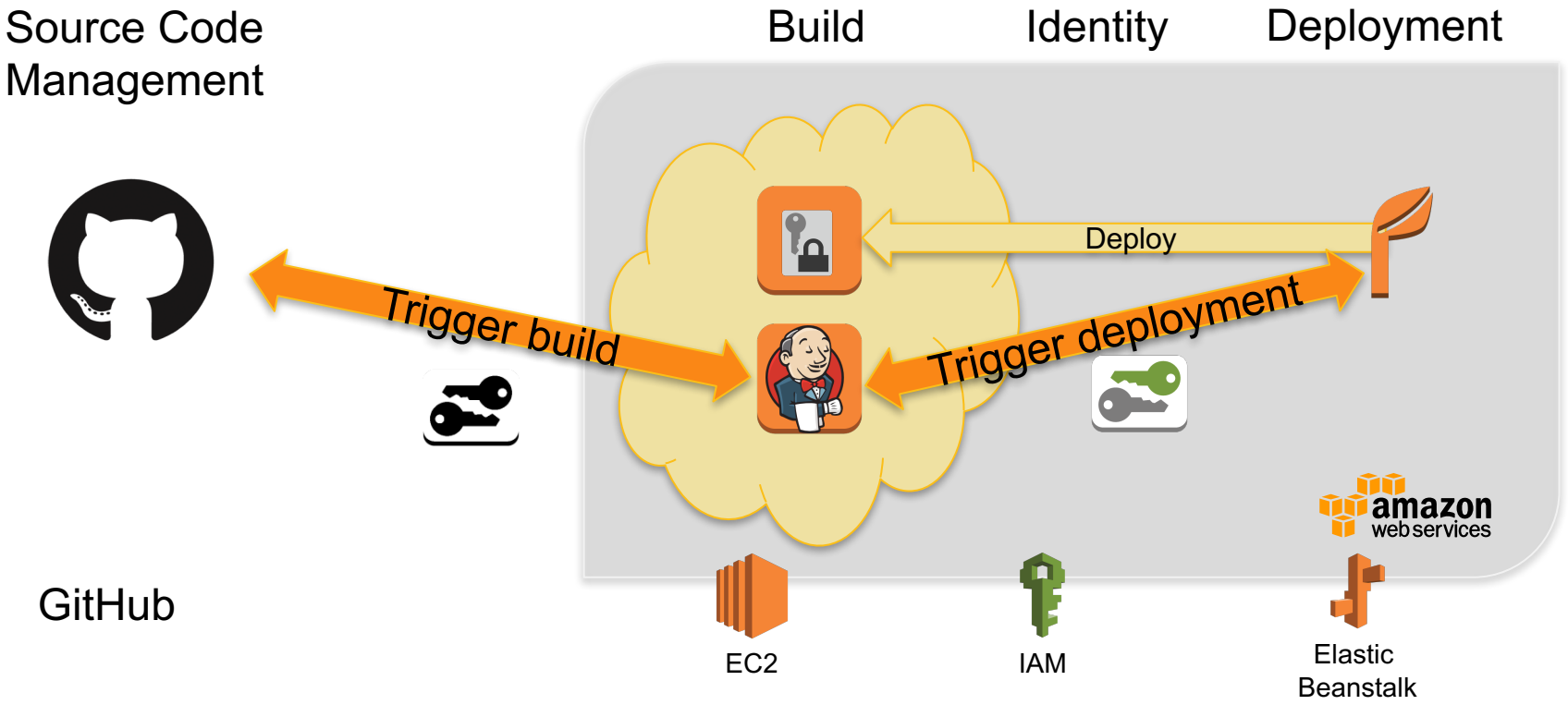
- A source code repository
- A centralized build server
- Deploy a simple micro service into the cloud

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

Code pipeline



Our Continuous Delivery Pipeline



Agenda

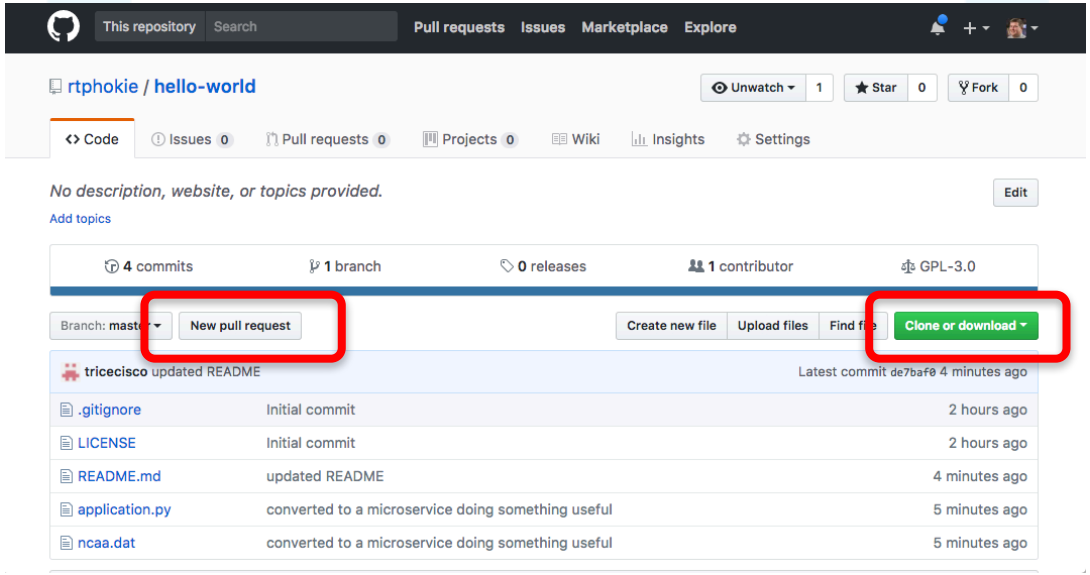
- 8:30-9:30 CI/CD overview / AWS and Git prerequisites
- 9:30-10:00 AWS Overview / Fork the GitHub Hello Word Repo
- 10:00:10:30 Provision EC2, Elastic Beanstalk and security groups
- 10:30-10:45 Break
- 10:45 - noon install Jenkins with Git and EB plugins
- break
- Noon-1:30 lunch
- 1:30 – 2:00 Vault overview
- 2-3:00 vault install and config
- 3-4 instrument Jenkins with vault secrets
- 4-5 build and deploy Hello World
- 5-5:30 wrap up and best practices

Cisco Public 2017 All Rights Reserved

Get the app source code from GitHub

- Visit `https://github.com/rtphokie/hello-world`

Pull …
then clone



Clone

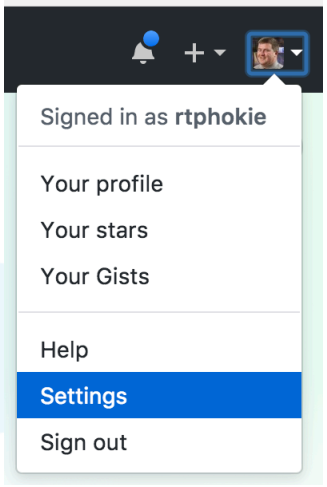
Join seccon_bootcamp organization

- https://github.com/seccon_bootcamp

- Or look for an invite in email

Create a GitHub developer key

Settings -> Developer Settings -> Personal access tokens



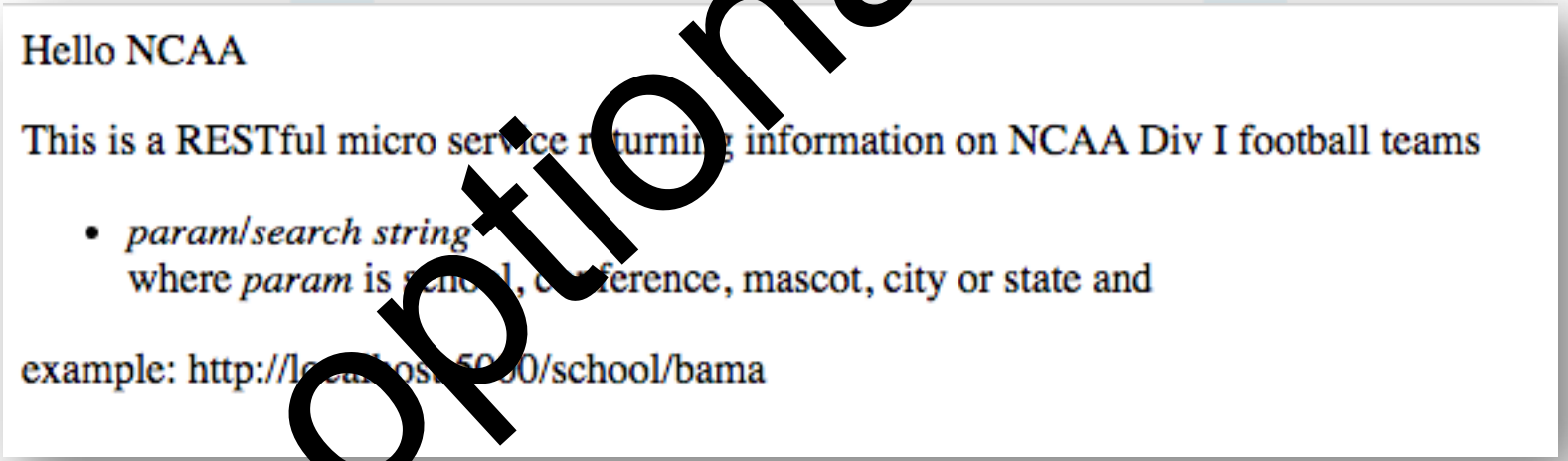
Get the app running locally

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

optional

Test application locally

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



optional

Our Continuous Delivery Pipeline

Source Code Management

GitHub

EC2

What is Vault?

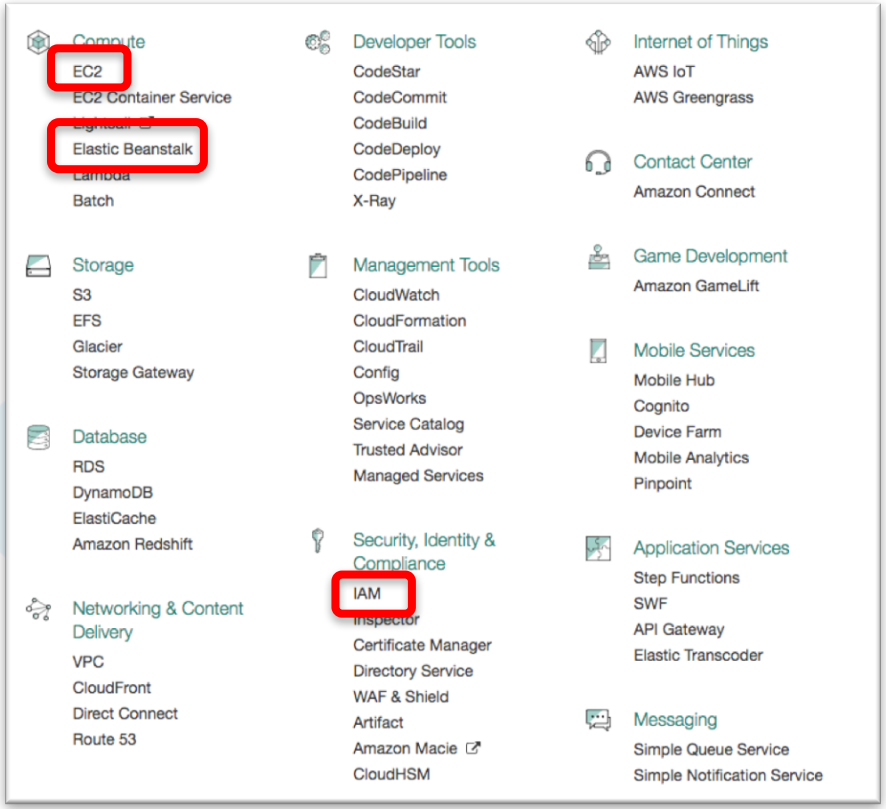
- A tool for Managing secrets.
- Software KMS (key management system).
- **Vault** secures, stores, and tightly controls access to secrets.

Agenda

- 8:30-9:30 CI/CD overview / AWS and Git prerequisites
- 9:30-10:00 AWS Overview / Fork the GitHub Hello Word Repo
- 10:00:10:30 Provision EC2, Elastic Beanstalk and security groups
- 10:30-10:45 Break
- 10:45 - noon install Jenkins with Git and EB plugins
- break
- Noon-1:30 lunch
- 1:30 – 2:00 Vault overview
- 2-3:00 vault install and config
- 3-4 instrument Jenkins with vault secrets
- 4-5 build and deploy Hello World
- 5-5:30 wrap up and best practices

Cisco Public 2017 All Rights Reserved

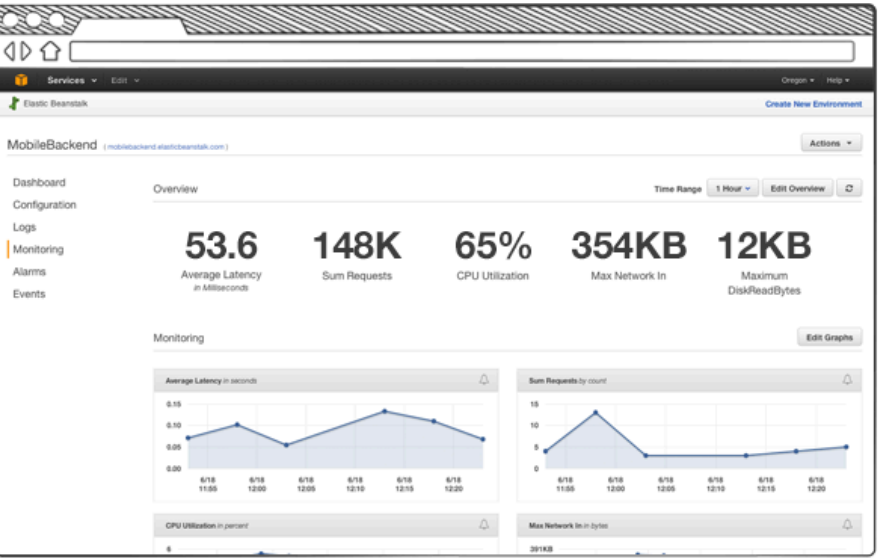
AWS console



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 — Looking for a different platform? Let us know.

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

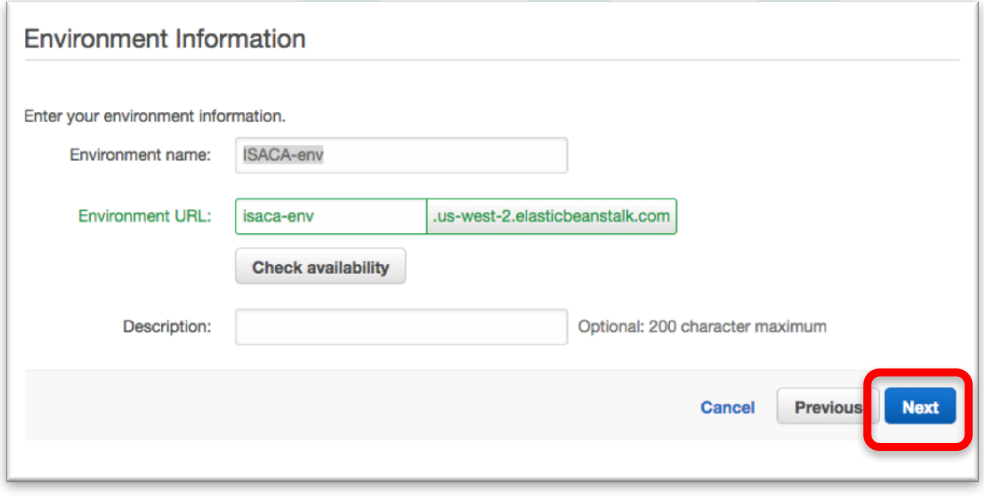
Launch Now

2.7 on 64 Bit Linux

CISCO

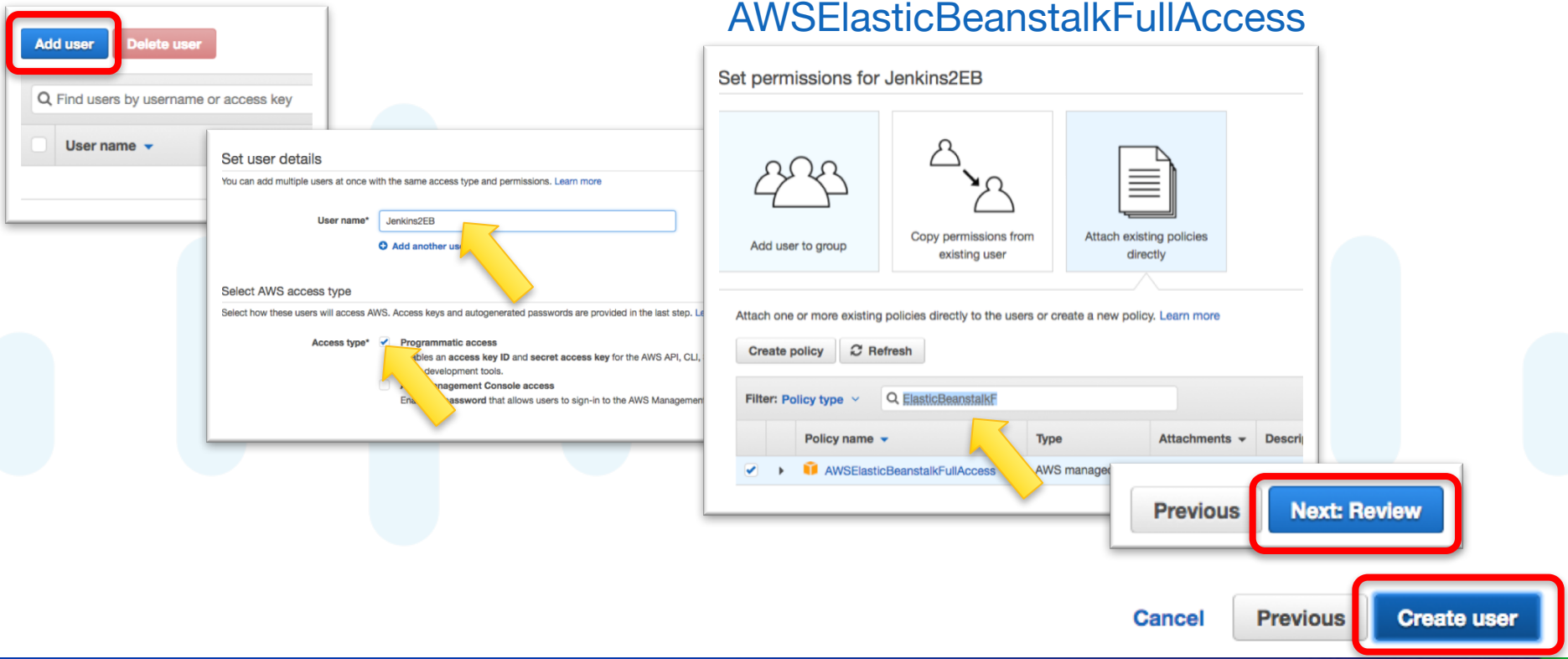
Elastic Beanstalk provision

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



IAM: Provision EB User

AWSElasticBeanstalkFullAccess



Our Continuous Delivery Pipeline

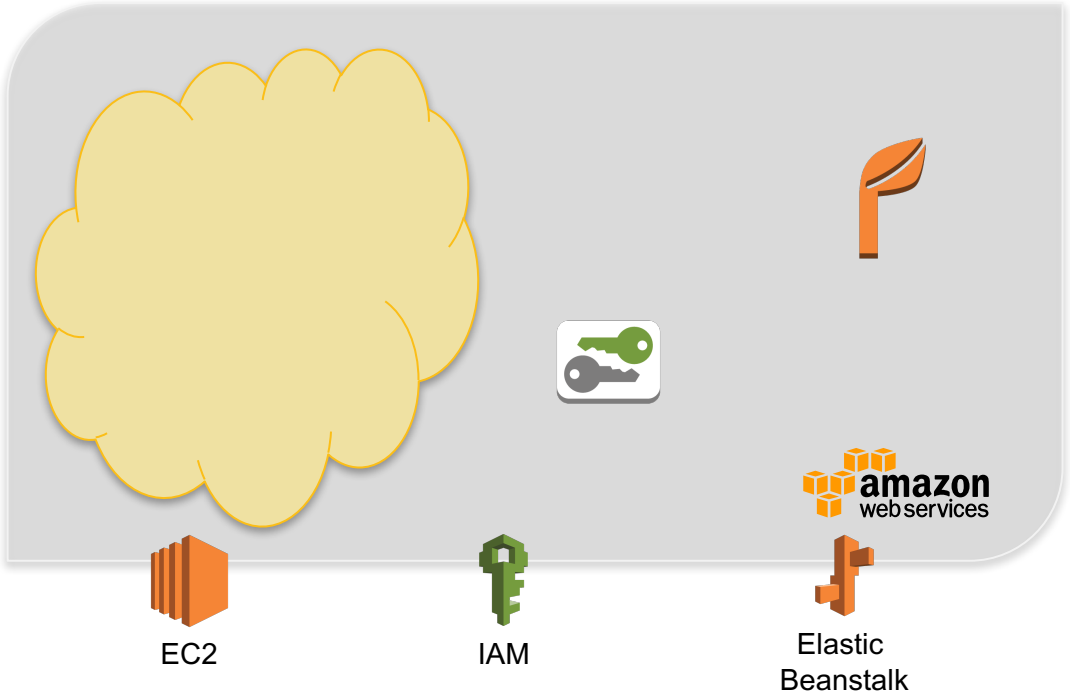
Source Code Management

GitHub

Build

Identity

Deployment

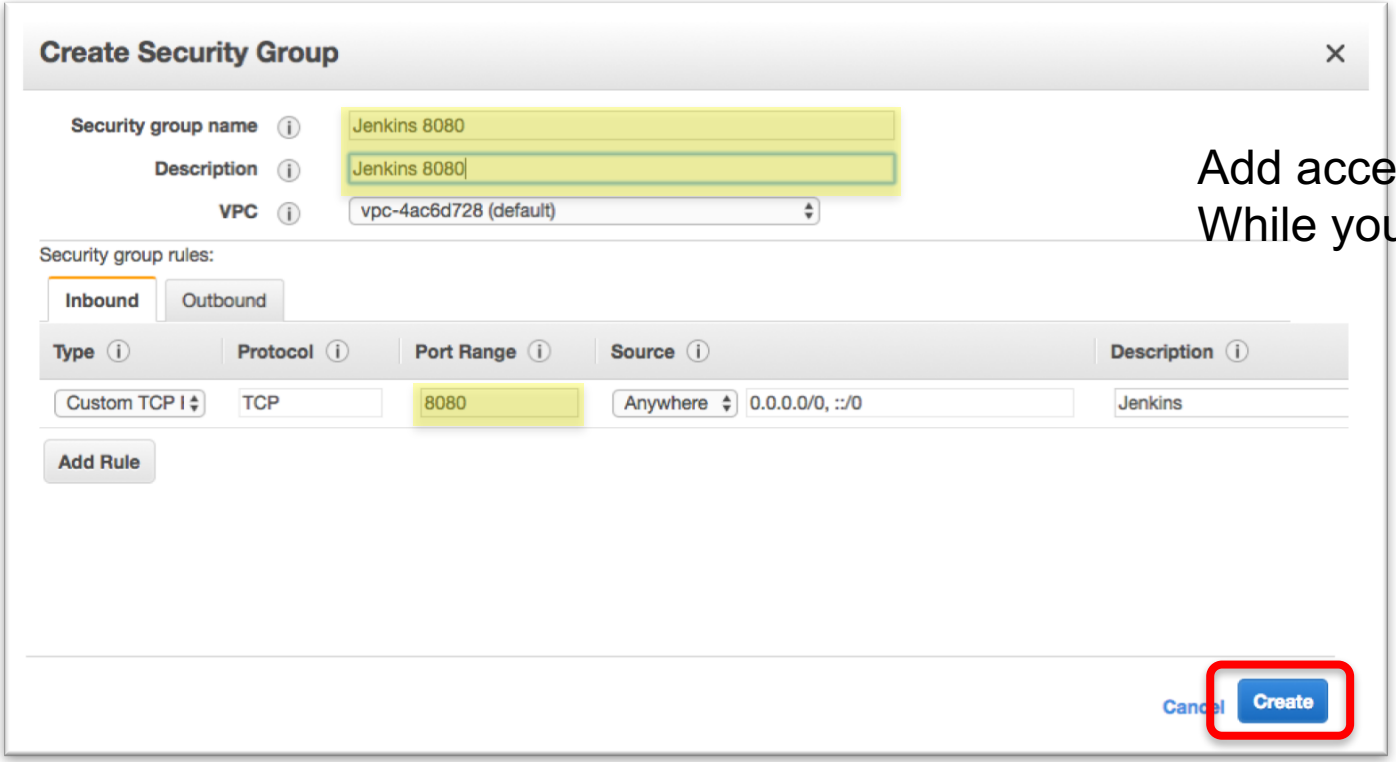


Agenda

- 8:30-9:30 CI/CD overview / AWS and Git prerequisites
- 9:30-10:00 AWS Overview / Fork the GitHub Hello Word Repo
- 10:00:10:30 Provision EC2, Elastic Beanstalk and security groups
- 10:30-10:45 Break
- 10:45 - noon install Jenkins with Git and EB plugins
- break
- Noon-1:30 lunch
- 1:30 – 2:00 Vault overview
- 2-3:00 vault install and config
- 3-4 instrument Jenkins with vault secrets
- 4-5 build and deploy Hello World
- 5-5:30 wrap up and best practices

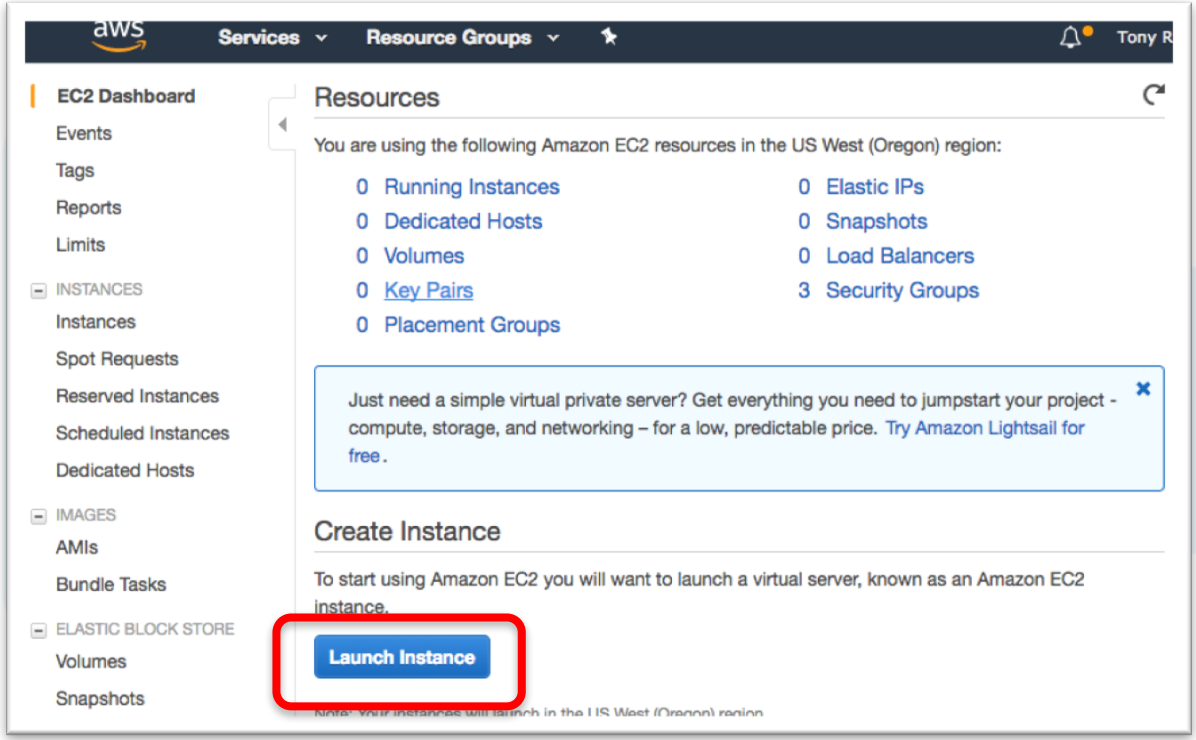
Cisco Public 2017 All Rights Reserved

EC2: security group

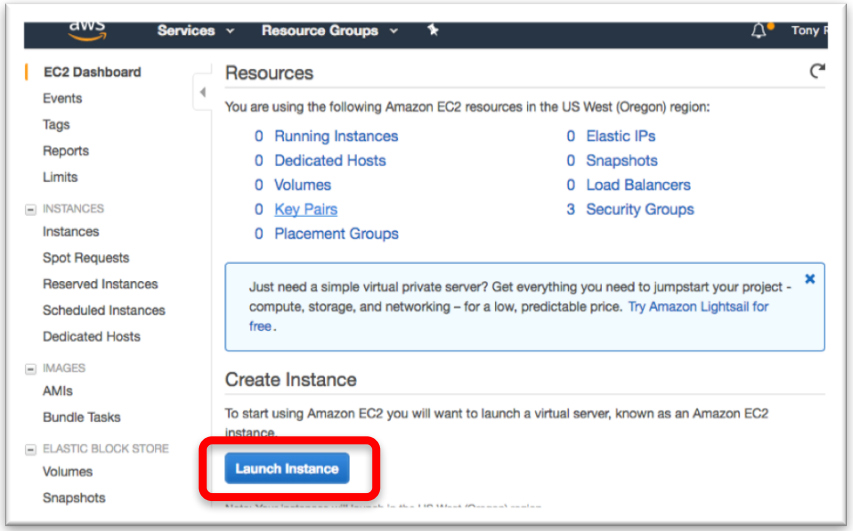


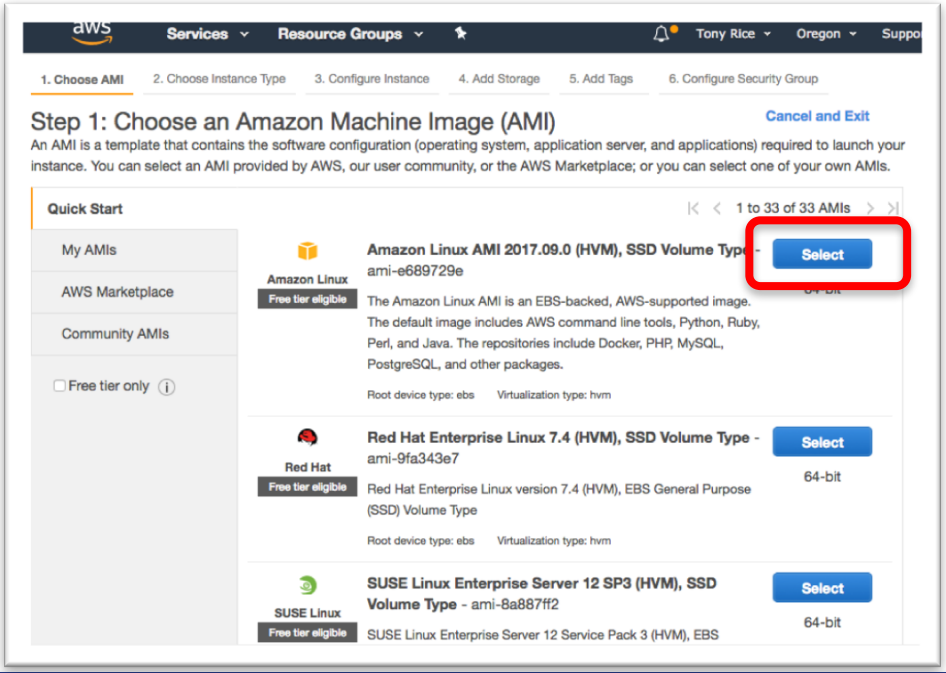
Add access to port 22
While you are creating this

EC2: launch EC2 instance

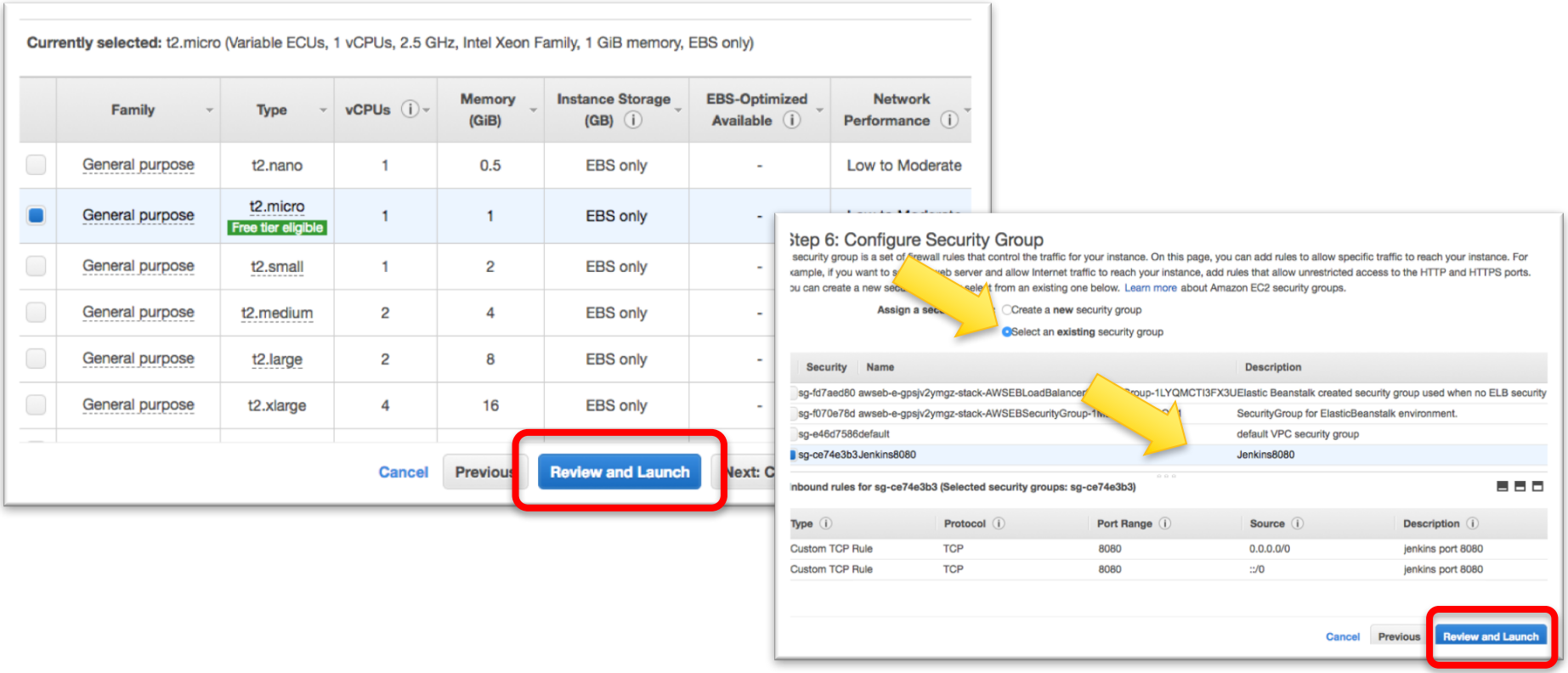


AWS: launch EC2 instance

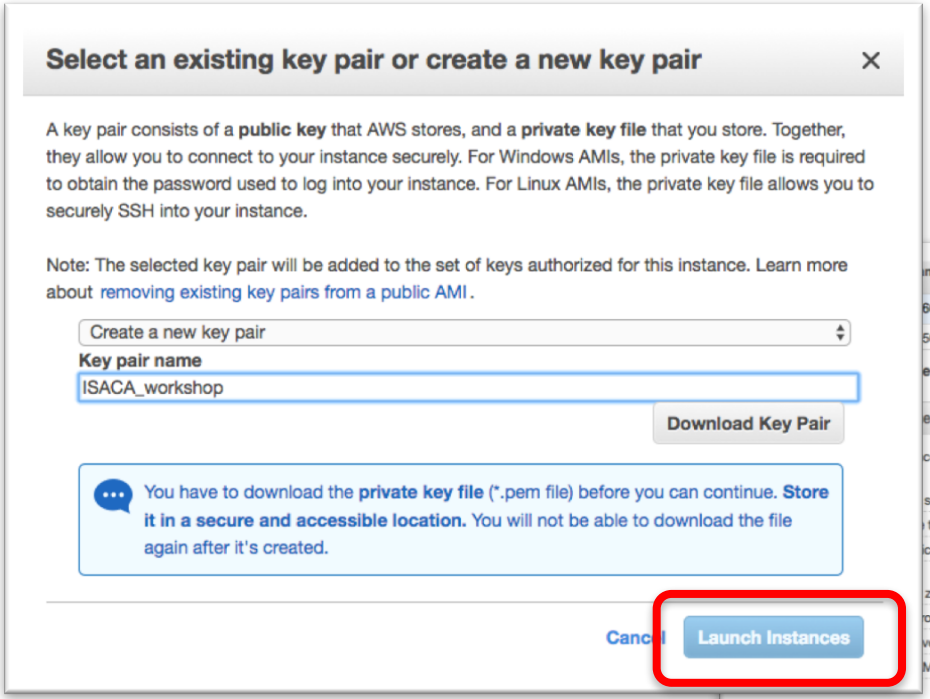




Select a free tier eligible instance type, launch



Create key pair & launch



Create new admin Linux user

- Add_user seconadmin
- Use Vault to generate and store key
- Log out of ec2-user
- Log in as seconadmin
  ssh –I <vaultgeneratedkey> seconadmin@<ip>

~/.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


       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|


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.
```

Agenda

- 8:30-9:30 CI/CD overview / AWS and Git prerequisites
- 9:30-10:00 AWS Overview / Fork the GitHub Hello Word Repo
- 10:00:10:30 Provision EC2, Elastic Beanstalk and security groups
- 10:30-10:45 Break
- 10:45 - noon install Jenkins with Git and EB plugins
- break
- Noon-1:30 lunch
- 1:30 – 2:00 Vault overview
- 2-3:00 vault install and config
- 3-4 instrument Jenkins with vault secrets
- 4-5 build and deploy Hello World
- 5-5:30 wrap up and best practices

Cisco Public 2017 All Rights Reserved

HashiCorp

Vault

A Tool for Managing Secrets

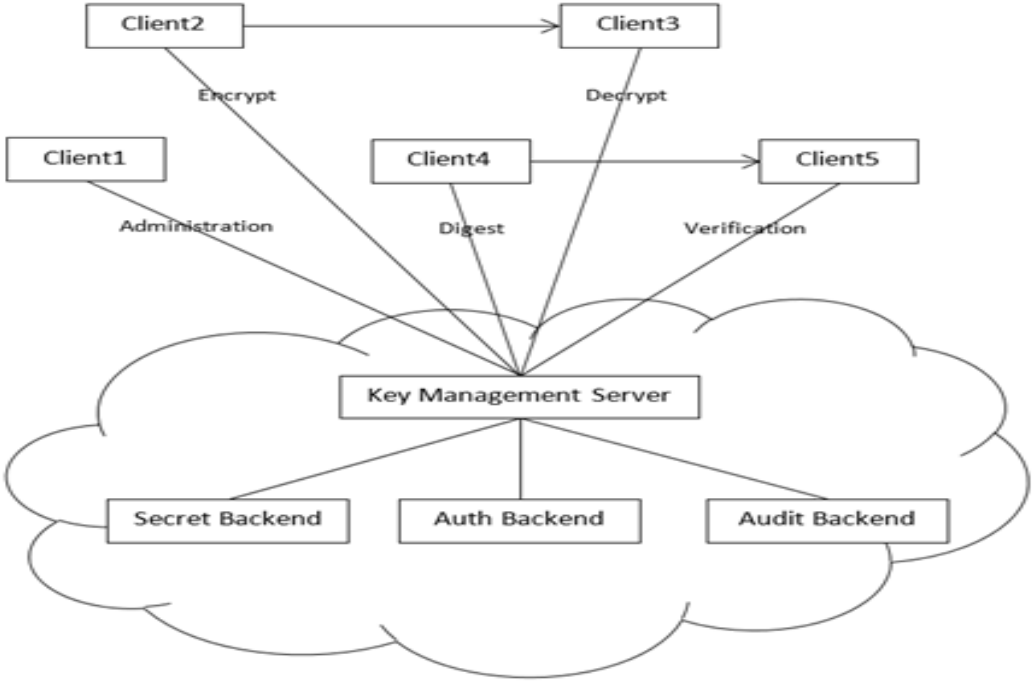
Why Vault?

- To leverage a single secure storage to store secrets such as API keys, passwords, certificates, and more for cloud applications.
- Unified interface, while ensuring strict controls and recording audit logs.
- Hardware HSMs though provides stronger controls are very expensive!!
- Ensures secrets are always stored separately from the data

*secret = passwords, tokens, private keys, API keys, symmetric encryption keys

Weak links in Cloud

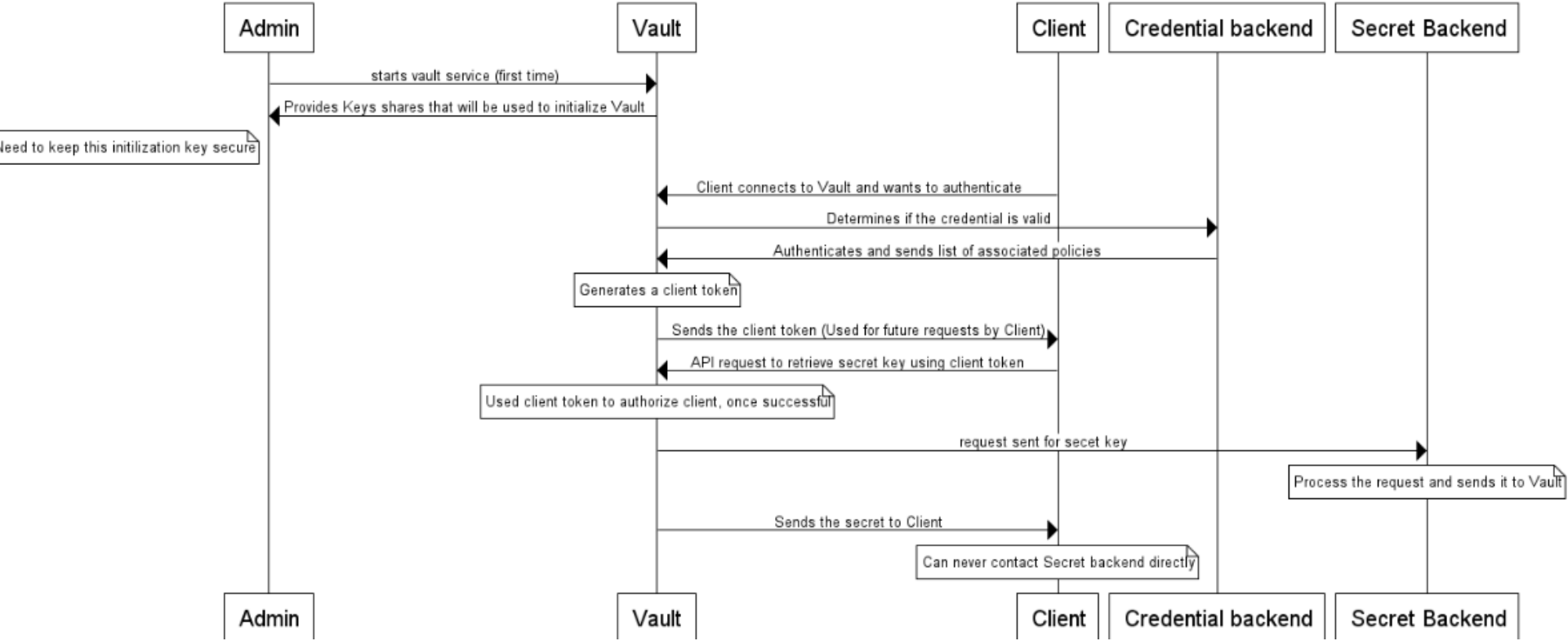
Key Management System overview



Key features of KMS

- Key generation
- Key Storage
- Key retrieval
- Policies for key rotation
- Manipulation of keys (activation date or description of key, revocation, delete)
- Access control or management
- Redundancy - mechanism for real time mirroring, load balancing, high availability
- Audit access - audit trail

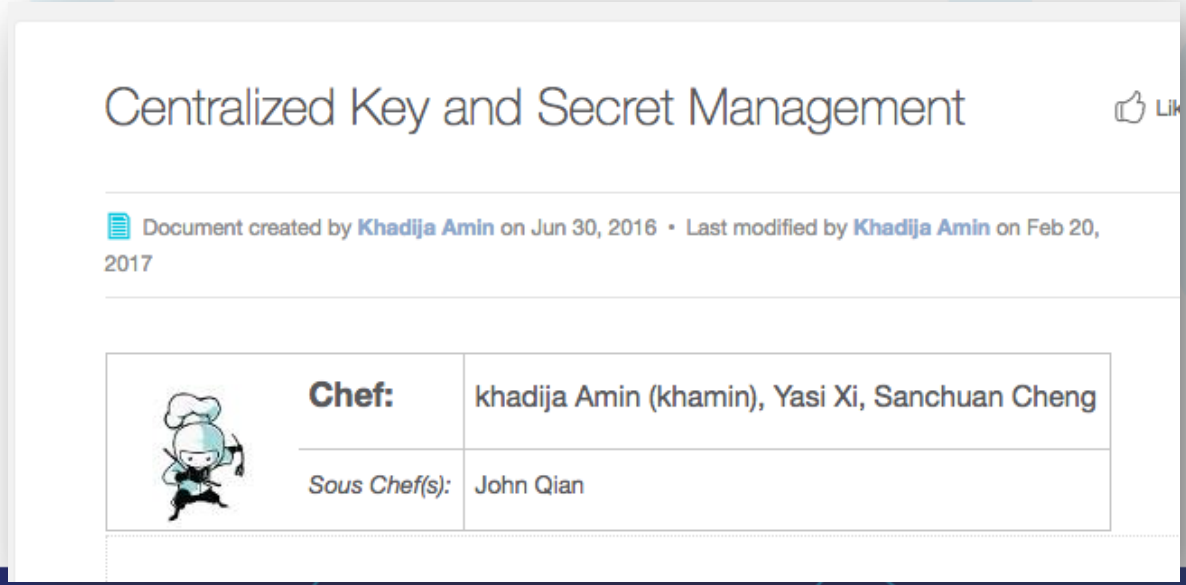
Vault high level example



Vault key strategies

- Deploy up-to-date vault version (v0.6.1 or above)
- Securely initialize and bring up the KMS
  - Protection of master keys that are needed to initialize the KMS
- Support High Availability
- Secure Communications Between Clients and KMS
- Authentication using automated ways (Tokens, PKI Keys)
- Roles and privileges should be well defined so that clients are granted least amount of privileges necessary to complete its task.

- More info on Vault can be found at the Infosec recipe – https://cisco.jiveon.com/docs/DOC-1510893



Setup vault

```
l9-5 ~]$
l9-5 ~]$ ./vault init
461169 [INFO ] core: security barrier not initialized
465330 [INFO ] core: security barrier initialized: shares=5 threshold=3
504942 [INFO ] core: post-unseal setup starting
523053 [INFO ] core: loaded wrapping token key
523210 [INFO ] core: successfully setup plugin catalog: plugin-directory=
529808 [INFO ] core: successfully mounted backend: type=kv path=secret/
529985 [INFO ] core: successfully mounted backend: type=cubbyhole path=cubbyhole/
530172 [INFO ] core: successfully mounted backend: type=system path=sys/
544246 [INFO ] expiration: restoring leases
544499 [INFO ] rollback: starting rollback manager
545759 [INFO ] expiration: lease restore complete
550890 [INFO ] core: post-unseal setup complete
551190 [INFO ] core/startClusterListener: starting listener: listener_address=127.0.0.1:8
551353 [INFO ] core/startClusterListener: serving cluster requests: cluster_listen_addres
558804 [INFO ] core: root token generated
558815 [INFO ] core: pre-seal teardown starting
558820 [INFO ] core: stopping cluster listeners
558825 [INFO ] core: shutting down forwarding rpc listeners
558840 [INFO ] core: forwarding rpc listeners stopped
951861 [INFO ] core: rpc listeners successfully shut down
951895 [INFO ] core: cluster listeners successfully shut down
951912 [INFO ] rollback: stopping rollback manager
951978 [INFO ] core: pre-seal teardown complete
5v8jiHc9H9p2N4uEVcAYDsp3bT3z0Z4HuamXQS
cXKJpbnSrHEg3Xub2/Bc1lYEXkGfC5tTSkaDi
svjs6MD+ZG0K1pjrWto/PHJ7bBbpRggSjXcEqd
z8nJeZ11vAQMLdX+fx56eLwDbdiAr6Zwov4Ler
T7/dhE38LionuHSCtdkZcCD+NJAXfFZ7YXeM5r
98e9ca13-5157-1543-6469-e24a9604be3a

th 5 keys and a key threshold of 3. Please
the above keys. When the vault is re-sealed,
d, you must provide at least 3 of these keys


 the master key. Without at least 3 keys,
in permanently sealed.
```

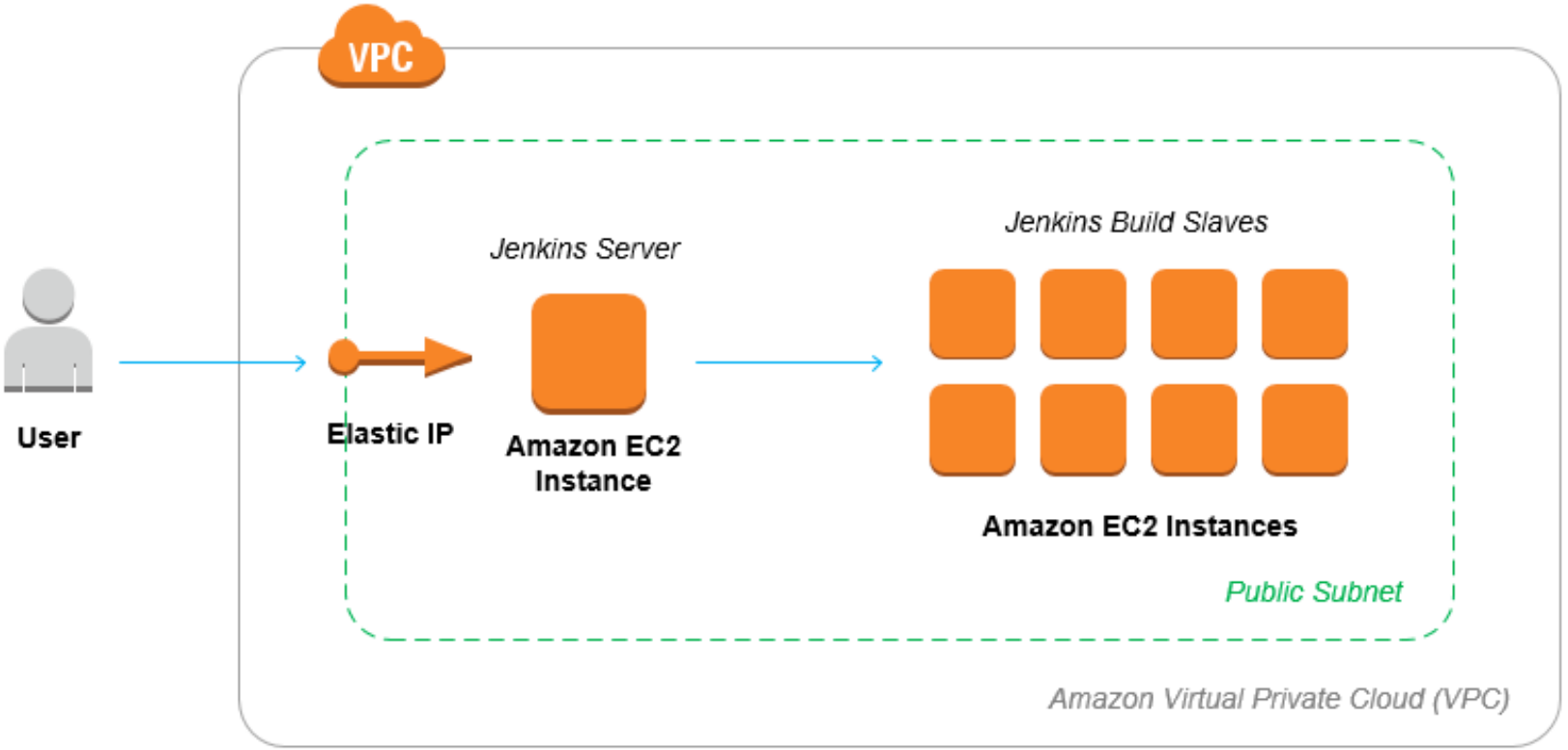
```
Vault is now sealed.
[ec2-user@ip-172-31-19-5 ~]$ ./vault status
Sealed: true
Key Shares: 5
Key Threshold: 3
Unseal Progress: 0
Unseal Nonce:
Version: 0.8.3

High-Availability Enabled: true
        Mode: sealed
[ec2-user@ip-172-31-19-5 ~]$ ./vault unseal
Key (will be hidden):
Sealed: true
Key Shares: 5
Key Threshold: 3
Unseal Progress: 1
Unseal Nonce: ae5f8f64-4162-d360-d267-9c05c6828f0a
[ec2-user@ip-172-31-19-5 ~]$ ./vault unseal
Key (will be hidden):
Sealed: true
Key Shares: 5
Key Threshold: 3
Unseal Progress: 2
Unseal Nonce: ae5f8f64-4162-d360-d267-9c05c6828f0a
[ec2-user@ip-172-31-19-5 ~]$ clear

[ec2-user@ip-172-31-19-5 ~]$ ./vault unseal
Key (will be hidden):
2017/10/31 21:06:45.106334 [INFO ] core: vault is unsealed
2017/10/31 21:06:45.106401 [INFO ] core: entering standby mode
    2017/10/31 21:06:45 [INFO] agent: Synced check 'vault:127.0.0.1:8200:vault-sealed-check'
Sealed: false
Key Shares: 5
Key Threshold: 3
Unseal Progress: 0
Unseal Nonce:
[ec2-user@ip-172-31-19-5 ~]$ 2017/10/31 21:06:45.115051 [INFO ] core: acquired lock, enabling a
2017/10/31 21:06:45.148488 [INFO ] core: post-unseal setup starting
2017/10/31 21:06:45.149037 [INFO ] core: loaded wrapping token key
2017/10/31 21:06:45.149046 [INFO ] core: successfully setup plugin catalog: plugin-directory=
2017/10/31 21:06:45.149649 [INFO ] core: successfully mounted backend: type=kv path=secret/
2017/10/31 21:06:45.149743 [INFO ] core: successfully mounted backend: type=system path=sys/
2017/10/31 21:06:45.149757 [INFO ] core: successfully mounted backend: type=cubbyhole path=cubb
```

Add a secret

From your Jenkins/Vault server

```
curl http://127.0.0.1:8200/v1/auth/github/login
  -d '{ "token": ”YourGitHubPersonalToken" }’

curl -H "X-Vault-Token: client_token"  -X POST  -d '{
"gittoken":”YourPersonalToken"}'
```

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  ]
```


https://d0.awsstatic.com/Projects/P5505030/aws-project_Jenkins-build-server.pdf

If you get Java8 errors

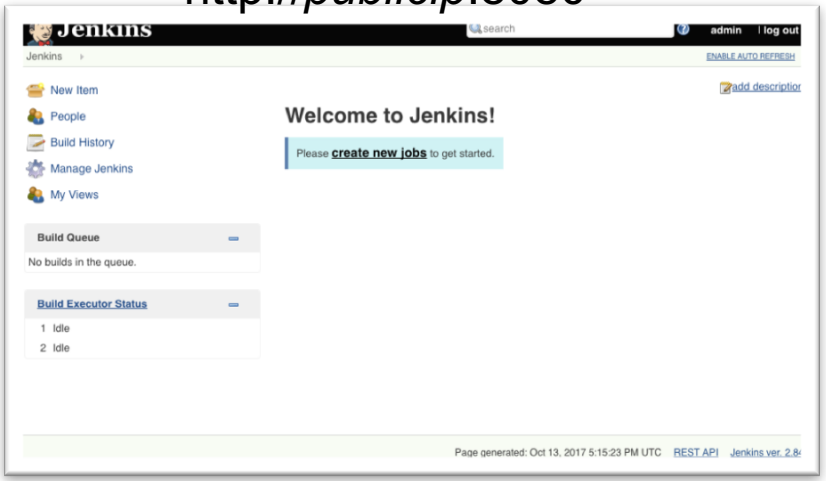
```
$ sudo service jenkins start
Starting Jenkins Jenkins requires Java8 or later, but you are running 1.7.0_151-
mockbuild_2017_08_09_21_42-b00 from /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.151.x86_64/jre
java.lang.UnsupportedClassVersionError: 51.0
at Main.main(Main.java:124)
$ sudo yum –Y install java-1.8.0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

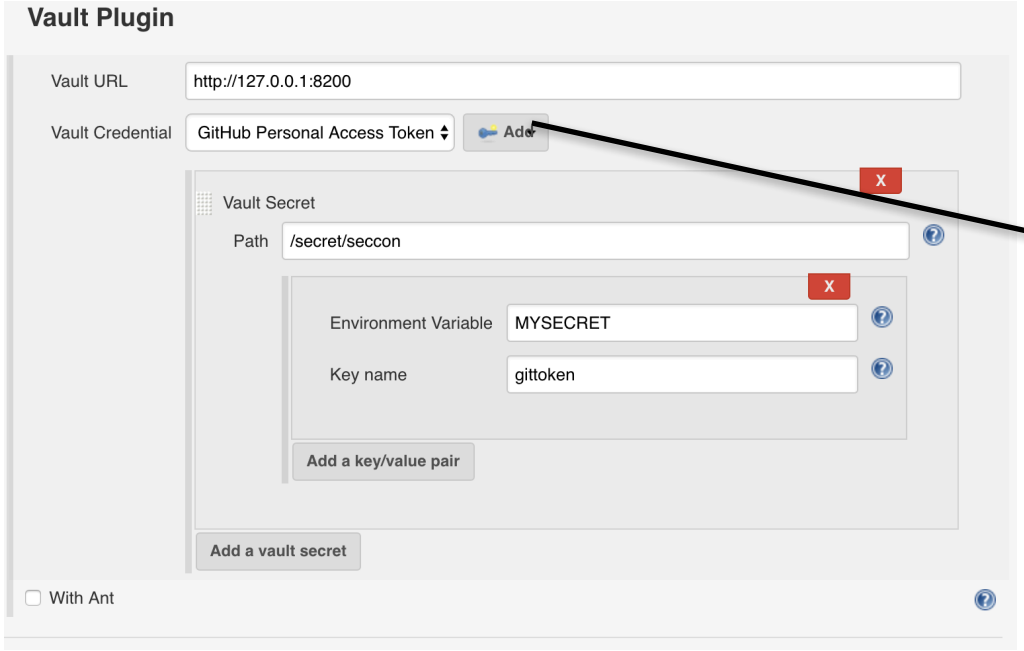


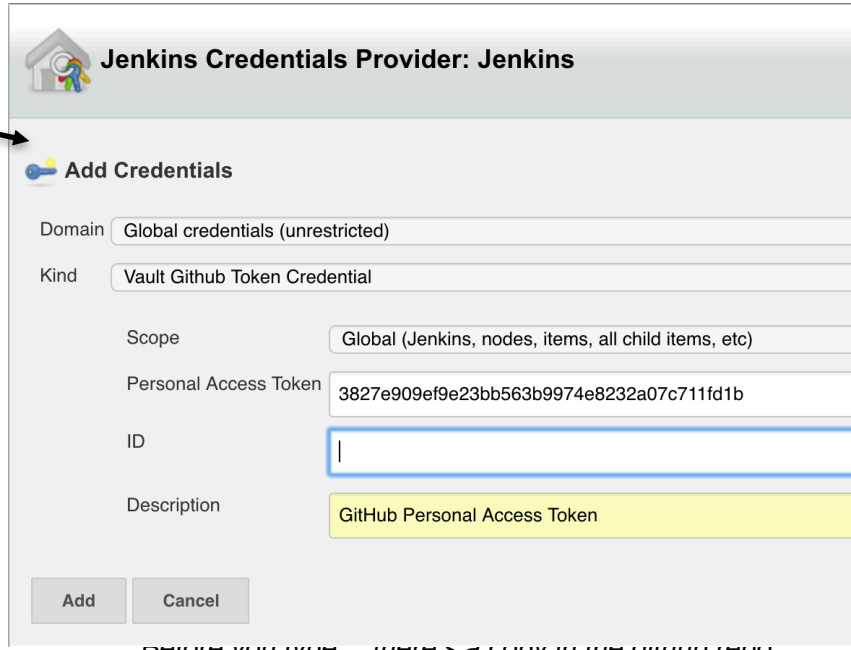
Create a build job – Embed vault secret

```
#!/bin/bash –x

VAULTTOKEN=(curl -s http://127.0.0.1:8200/v1/auth/github/login -d '{ "token": ”YOURVAULTTOKEN" }' |
python -c 'import sys, json; print json.load(sys.stdin)["auth"]["client_token"]')

GITTOKEN=(curl -s -H "X-Vault-Token: $VAULTTOKEN” -X GET http://127.0.0.1:8200/v1/secret/seccon|
python -c 'import sys, json; print json.load(sys.stdin)["data"]["gittoken"]')

if [ -d "hello-world" ];
then
cd hello-world
git pull
else
git clone https://$GITTOKEN:x-oauth-basic@github.com/rtphokie/hello-world.git
Fi
rm -Rf hello-world
```

Before you type… there's a copy in the github repo

Create a build job – Vault Plugin





Before you type… there's a copy in the github repo

Create a build job – Use secret to fetch code

```
#!/bin/bash –x

if [ -d "hello-world" ]; then
        cd hello-world;
        git pull
else
        url = https://$MYSECRET:x-oauth-basic@github.com/rtphokie/hello-world.git
        git clone $URL
fi
rm -Rf hello-world
```

Before you type… there's a copy in the github repo

Our Continuous Delivery Pipeline

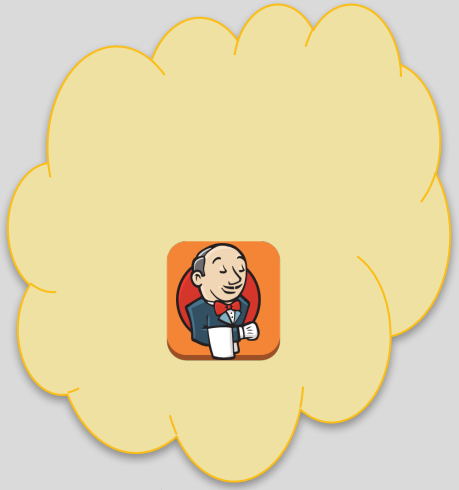
Source Code Management

GitHub

Build

Identity

Deployment



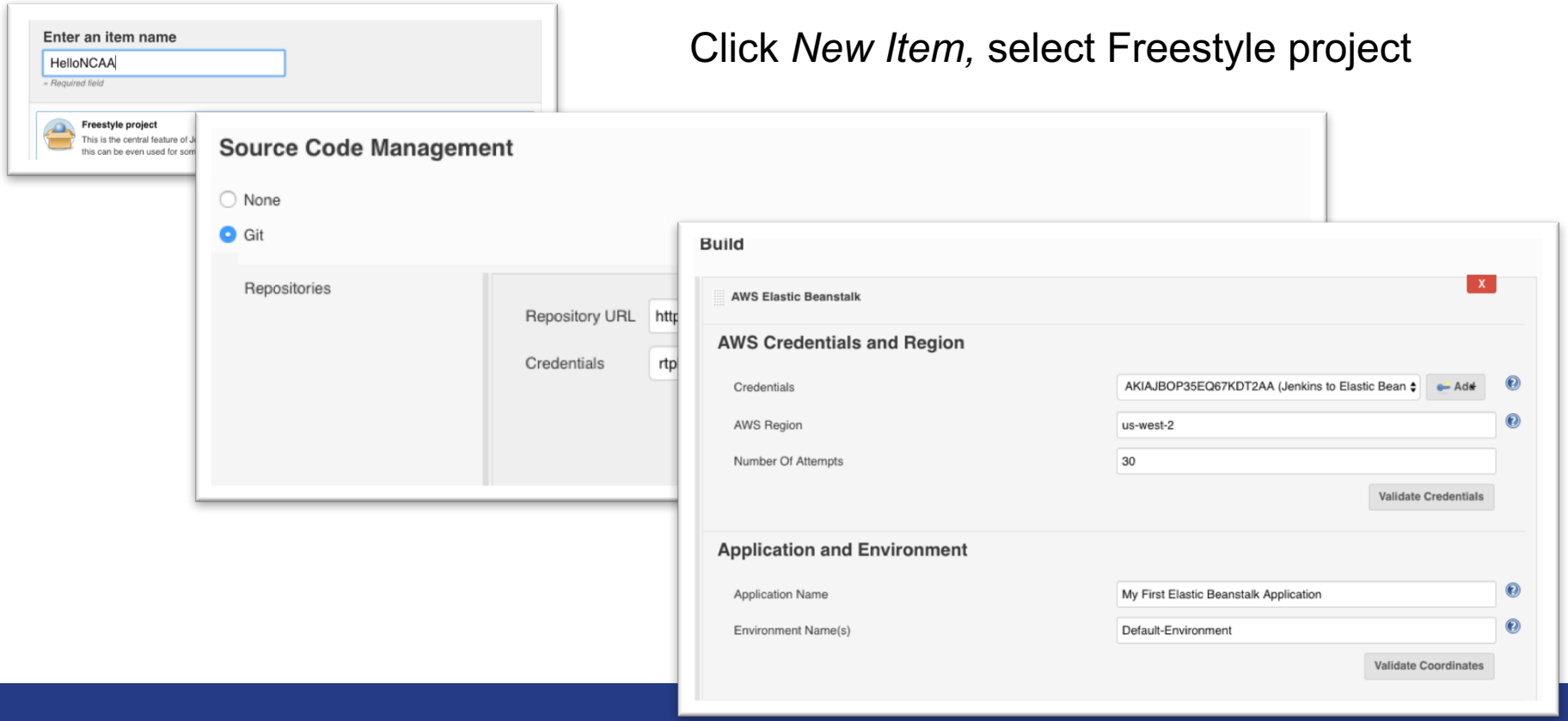
EC2

Elastic Beanstalk

Create a key pair (Github – Jenkins)

- Login to EC2 instance
- Switch to the Jenkins user
  `sudo su – jenkins`
- Generate a key
  `ssh-keygen -t rsa`
- Public and private is_rsa files are in ~jenkins/.ssh

From the Jenkins GUI: Create a build job



Click *New Item,* select Freestyle project

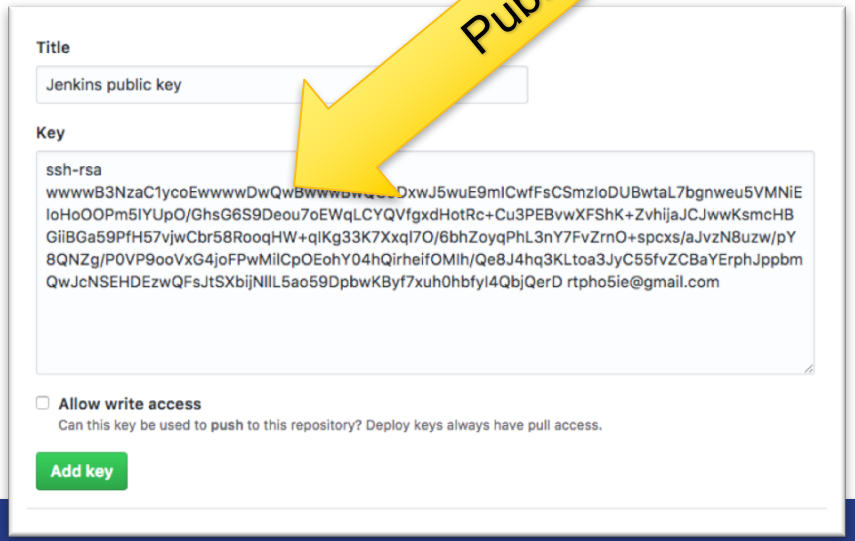
Github: Deploy key

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

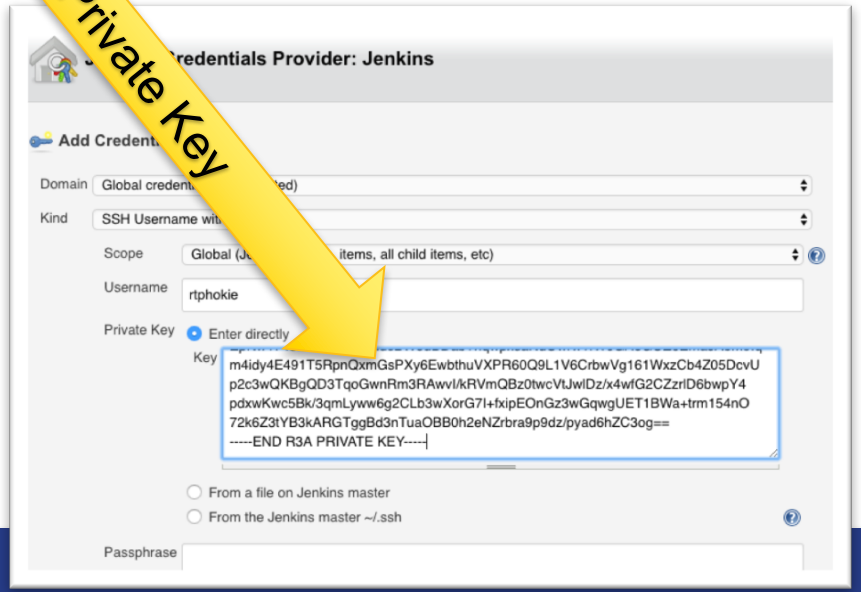
Public key

Private Key

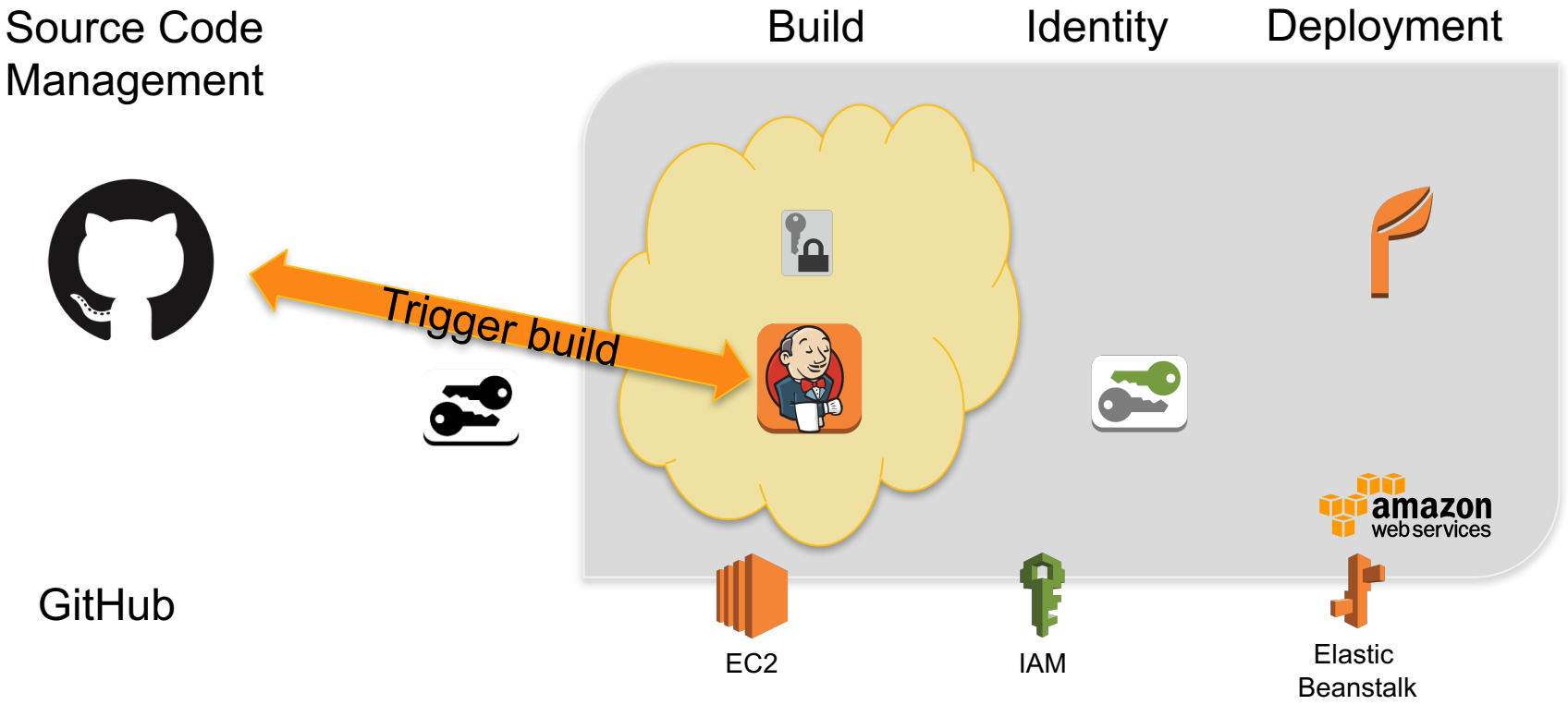
Github



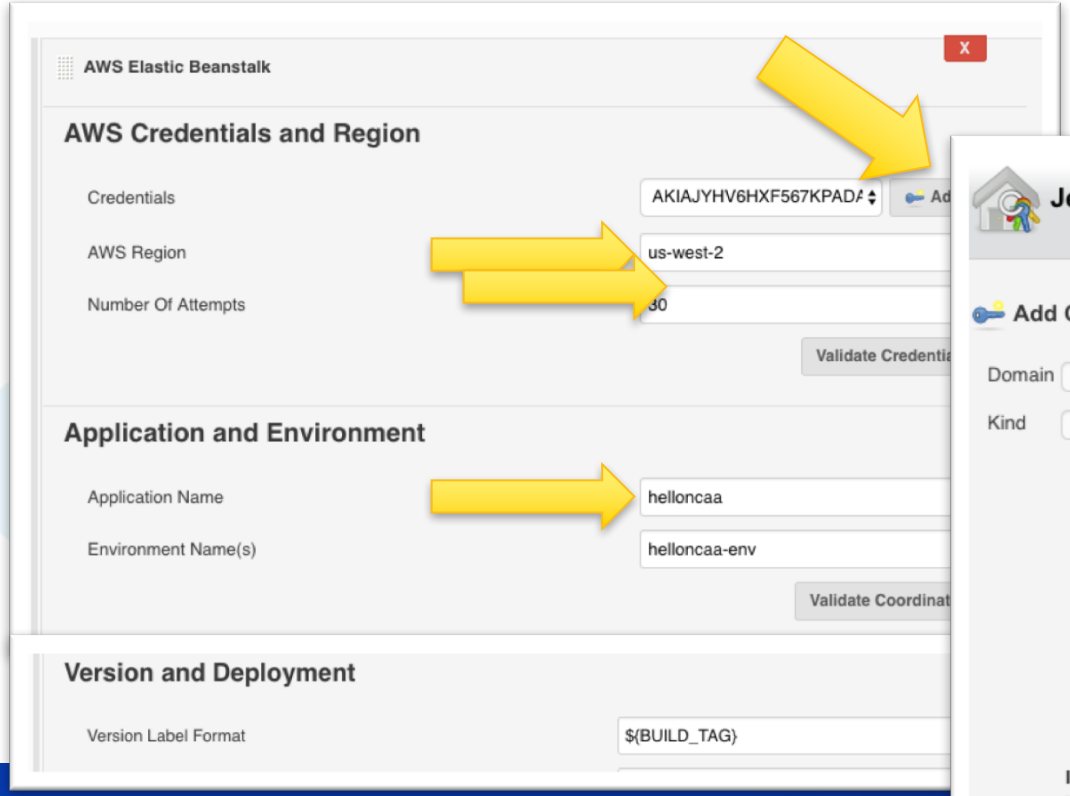
Jenkins



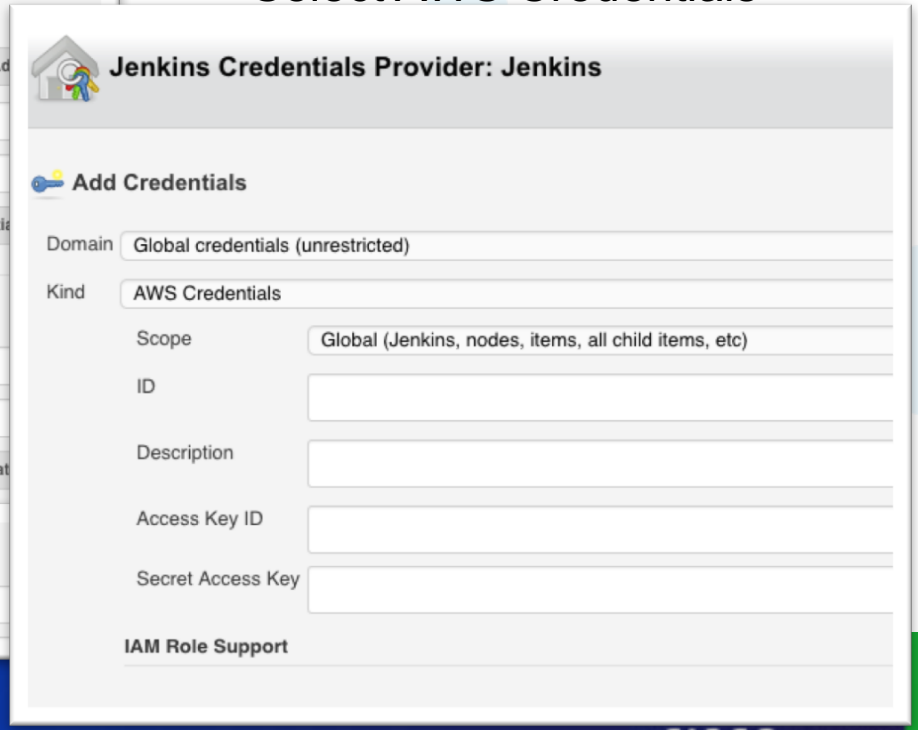
Our Continuous Delivery Pipeline



Jenkins: deploy to EB post build

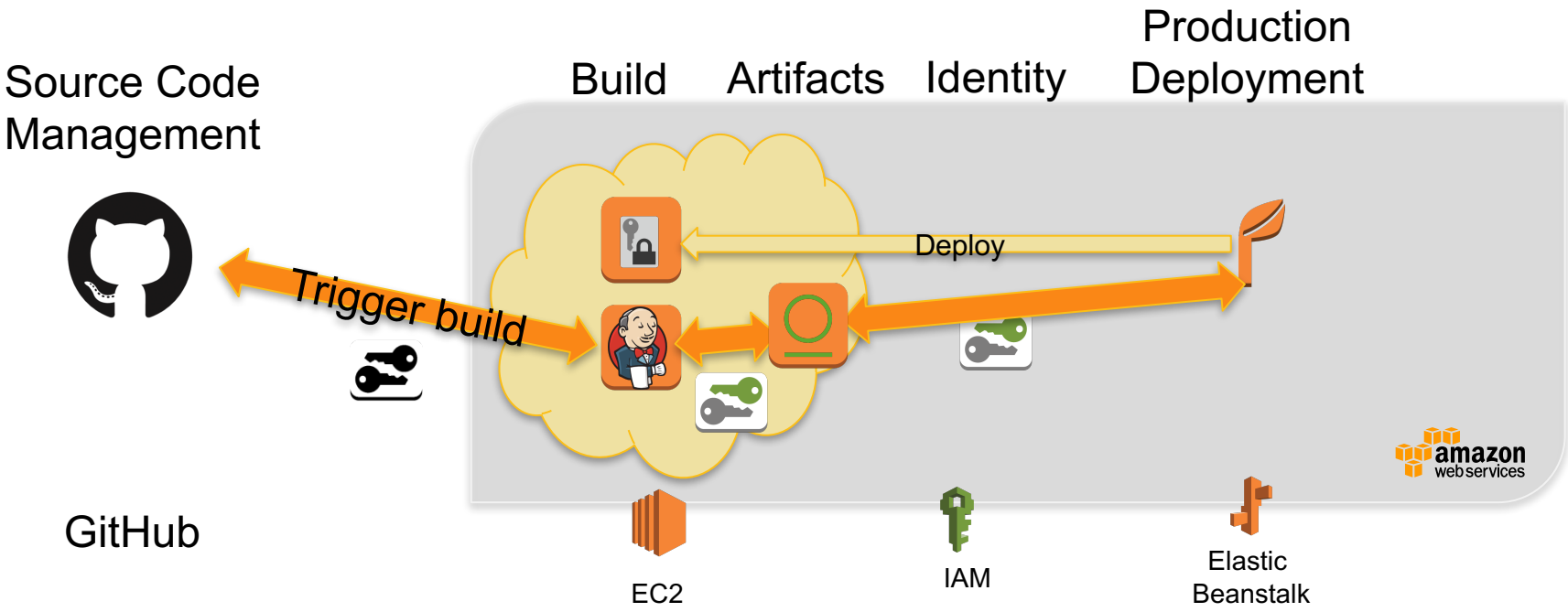


Select AWS Credentials

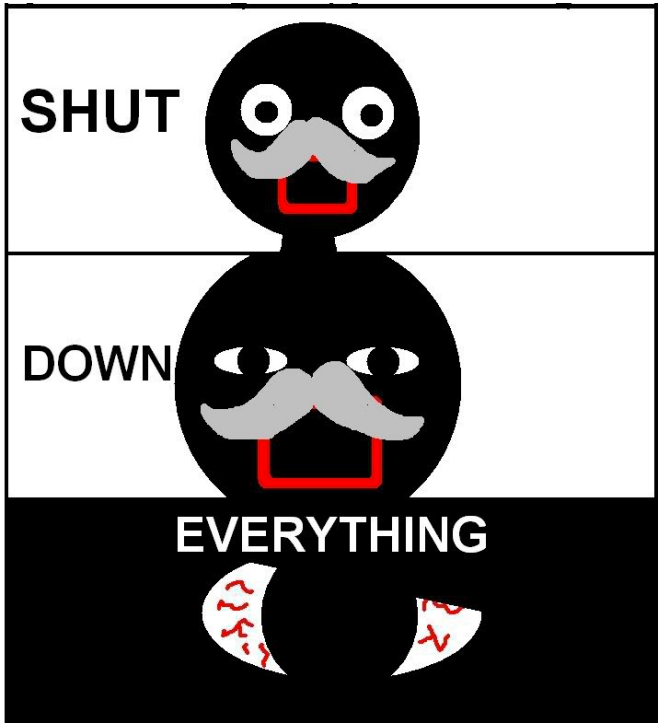


CISCO

Artifact Management



Before you go



Avoid unexpected bills:

Terminate all EC2 instances and Elastic Beanstalk apps.

Thank you.

trice@cisco.com

@rtphokie