

# Testing in DevOps

DOu – Certified Tester in DevOps (CTD)

Exercise Solutions







# Setup Code

Expense Manager, System Test, BDD Test, Feature Toggle Code





# Step 1 - Getting Sample Code(Expense Manager App, System Test)

- Open the browser and go to <a href="https://github.com/login">https://github.com/login</a>
- Login to your account
- Launch URL <a href="https://github.com/umangsaltuniv/verity-devops-ex">https://github.com/umangsaltuniv/verity-devops-ex</a>
- Click Fork at right top section
- "verity-devops-ex" repository will be added on your GitHub account
- Launch URL <a href="https://github.com/umangsaltuniv/EMSystemTests-ex">https://github.com/umangsaltuniv/EMSystemTests-ex</a>
- Click Fork at right top section
- "EMSystemsTests-ex" repository will be added on your GitHub account

Note: verity-devops-ex project has Expense Manager sample app & some unit tests those will run on the app to do unit testing of the app. EMSystemTests-ex project has system test that will run on the app to do system testing of the app.



# Step 1 - Getting Sample Code(BDD Test, API Test)

- Launch URL <a href="https://github.com/umangsaltuniv/EMSystemTests">https://github.com/umangsaltuniv/EMSystemTests</a> BDD
- Click Fork at right top section
- "EMSystemTests\_BDD" repository will be added on your GitHub account
- Launch URL <a href="https://github.com/umangsaltuniv/EMAPITests-ex">https://github.com/umangsaltuniv/EMAPITests-ex</a>
- Click Fork at right top section
- "EMAPITests-ex" repository will be added on your GitHub account

Note: EMSystemTests\_BDD project has bdd test that will run on the app to do user acceptance testing. EMAPITests-ex project has api test that will run on the app to do api testing.



# Step 2 - Cloning Sample Code(Expense Manager App)

- After forking "verity-devops-ex" repository on GitHub Web, Click "Code" button
- Click "Open with GitHub Desktop"
- Click "Open GitHubDesktop"
- GitHub Desktop UI will be launched
- Browser C:\DO-United\GitHubRepo path under "Local path" section
- Click Clone
- GitHub Desktop UI will open the cloned repository and code will be downloaded on local machine under C:\DO-United\GitHubRepo



# Step 3 - Cloning Sample Code(System Tests)

- After forking "EMSystemTests-ex" repository on GitHub Web, Click "Code" button
- Click "Open with GitHub Desktop"
- Click "Open GitHubDesktop"
- GitHub Desktop UI will be launched
- Browser C:\DO-United\GitHubRepo path under "Local path" section
- Click Clone
- GitHub Desktop UI will open the cloned repository and code will be downloaded on local machine under C:\DO-United\GitHubRepo





# Step 4 - Cloning Sample Code(BDD Tests)

- After forking "EMSystemTests\_BDD" repository on GitHub Web, Click "Code" button
- Click "Open with GitHub Desktop"
- Click "Open GitHubDesktop"
- GitHub Desktop UI will be launched
- Browser C:\DO-United\GitHubRepo path under "Local path" section
- Click Clone
- GitHub Desktop UI will open the cloned repository and code will be downloaded on local machine under C:\DO-United\GitHubRepo



# Step 5 - Cloning Sample Code(API Test)

- After forking "EMAPITests-ex" repository on GitHub Web, Click "Code" button
- Click "Open with GitHub Desktop"
- Click "Open GitHubDesktop"
- GitHub Desktop UI will be launched
- Browser C:\DO-United\GitHubRepo path under "Local path" section
- Click Clone
- GitHub Desktop UI will open the cloned repository and code will be downloaded on local machine under C:\DO-United\GitHubRepo





# Setup Jenkins on AWS





# Setup Jenkins on AWS

- Setup AWS Ubuntu machine
- Setup Docker
- Setup Jenkins
- Setup Jenkins project
- Setup Webhook in GitHub web & Jenkins project

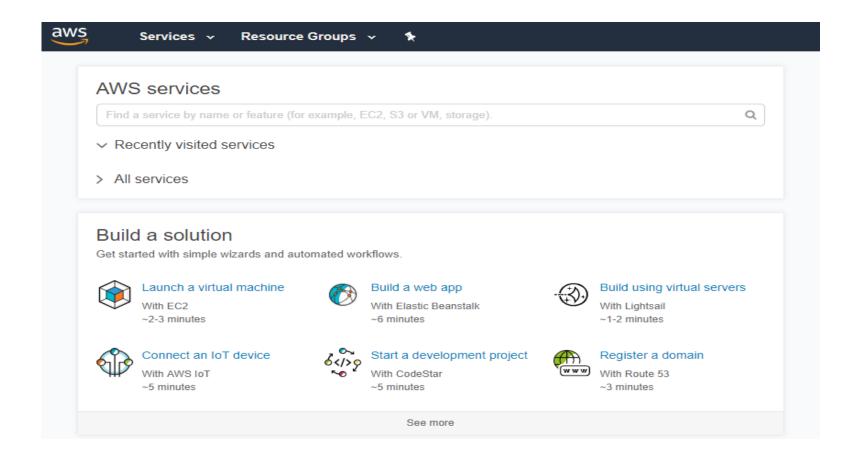




# Setup AWS Ubuntu machine



# Login to AWS & Navigate to Home Page





# Invoking EC2

Find a service by name or feature (for example, EC2, S3 or VM, storage). Group A-Z Compute **Management Tools** Security, Identity & **Desktop & App Streaming** Compliance EC2 CloudWatch WorkSpaces IAM Lightsail @ AppStream 2.0 AWS Auto Scaling **ECS** Cognito CloudFormation Secrets Manager **EKS** CloudTrail **Internet Of Things** GuardDuty Lambda Config Inspector IoT Core Batch OpsWorks Elastic Beanstalk Service Catalog Amazon Macie 2 IoT 1-Click Systems Manager **AWS Organizations** IoT Device Management AWS Single Sign-On Trusted Advisor IoT Analytics Storage Certificate Manager Greengrass Managed Services Key Management Service S3 Amazon FreeRTOS CloudHSM **EFS** IoT Device Defender Media Services S3 Glacier Directory Service WAF & Shield Elastic Transcoder Storage Gateway **Game Development** Kinesis Video Streams Artifact MediaConvert Amazon GameLift Database MediaLive Mobile Services RDS MediaPackage Mobile Hub DynamoDB MediaStore

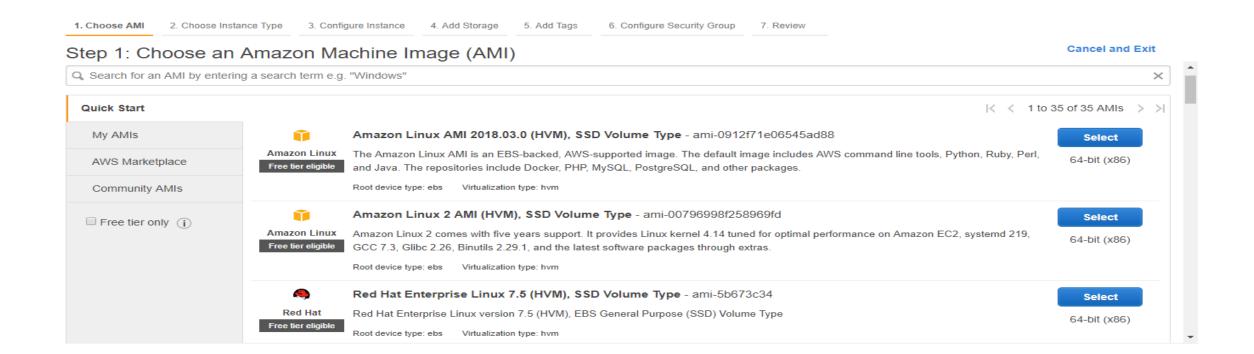


## EC2 Dashboard

 Click Launch Instance EC2 Dashboard Resources Events You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) region: Tags 0 Running Instances 0 Elastic IPs Reports 0 Dedicated Hosts 0 Snapshots Limits 0 Volumes 0 Load Balancers 1 Security Groups 0 Key Pairs INSTANCES Instances 0 Placement Groups Launch Templates Create Instance Spot Requests Reserved Instances To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance. Dedicated Hosts Launch Instance Capacity Reservations Note: Your instances will launch in the Asia Pacific (Mumbai) region IMAGES Service Health Scheduled Events AMIs **Bundle Tasks** Service Status: Asia Pacific (Mumbai): No events Asia Pacific (Mumbai): ELASTIC BLOCK STORE Availability Zone Status: Volumes ap-south-1a: Snapshots Availability zone is operating normally Lifecycle Manager



# Choosing an AMI





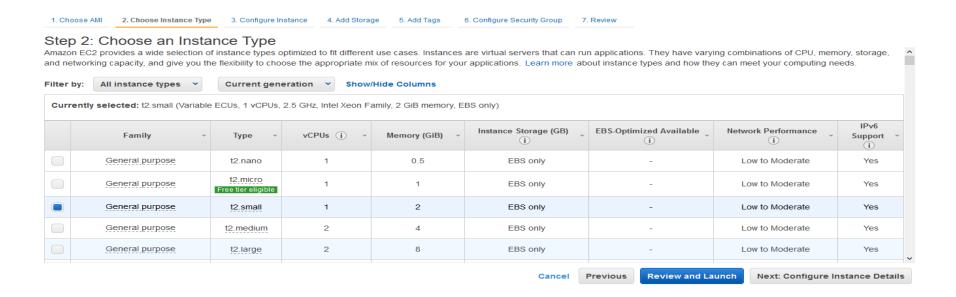
# Choosing Ubuntu Image(Ubuntu Server 18.04) Do not use any other image

Click Select **6** Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0d773a3b7bb2bb1c1 Select Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical 64-bit (x86) Free tier eligible (http://www.ubuntu.com/cloud/services). Root device type: ebs Virtualization type: hvm



# Choosing Instance Type(t2.small)

Click Next: Configure Instance Details





# Configuring Instance Details(Keep as default)

 Click Next: Add Storage 1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group Step 3: Configure Instance Details Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more. Number of instances (i) Launch into Auto Scaling Group (i) Purchasing option (i) Request Spot instances Network (i) vpc-dd0b2db5 (default) ▼ C Create new VPC Subnet (i) No preference (default subnet in any Availability Zon₁ ▼ Auto-assign Public IP (i) Use subnet setting (Enable) Placement group (i) Add instance to placement group. Capacity Reservation (i) ▼ C Create new Capacity Reservation IAM role (i) ▼ C Create new IAM role Shutdown behavior (i) **Review and Launch** Next: Add Storage



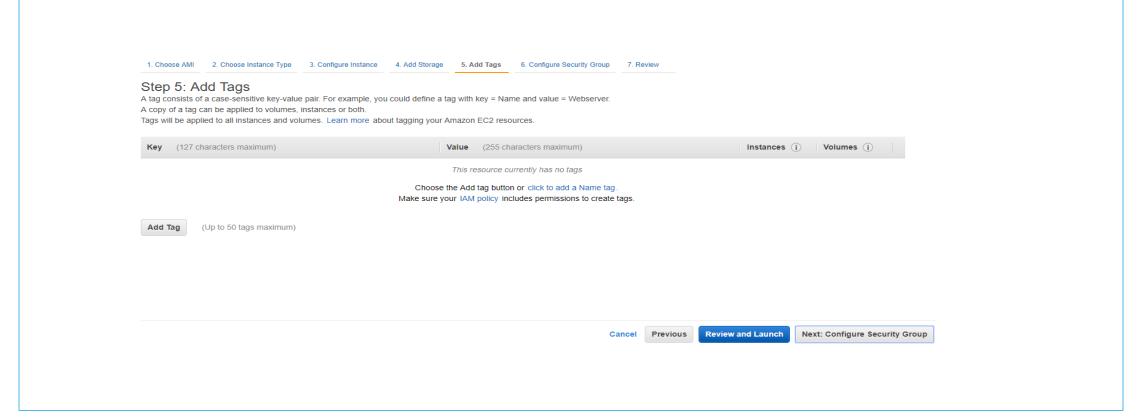
# Adding Storage(Keep as default)

 Click Next: Add Tags 1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags Step 4: Add Storage Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2. Delete on Volume Type (i) Size (GiB) (i Volume Type (i) Encrypted (i) /dev/sda1 snap-02fb12e5e1d2255d6 General Purpose SSD (gp2) Not Encrypted Add New Volume Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and **Review and Launch** Next: Add Tags Previous



# Adding Tags(Keep as default)

Click Next: Configure Security Group





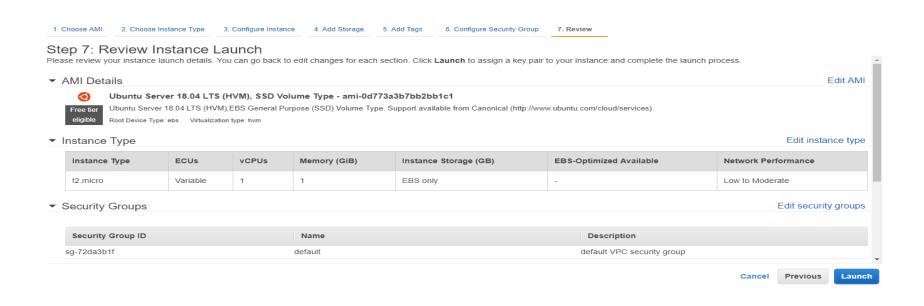
# Configuring Security Group(Keep as default)

 Click Review and Launch 1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure 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: Oreate a new security group Select an existing security group Security Group ID Actions Name Description sq-72da3b1f default VPC security group Copy to new Inbound rules for sg-72da3b1f (Selected security groups: sg-72da3b1f) Type (i) Protocol (i) Port Range (i) Source (i) Description (i) All traffic sg-72da3b1f (default) Cancel Previous Review and Launch



# Reviewing

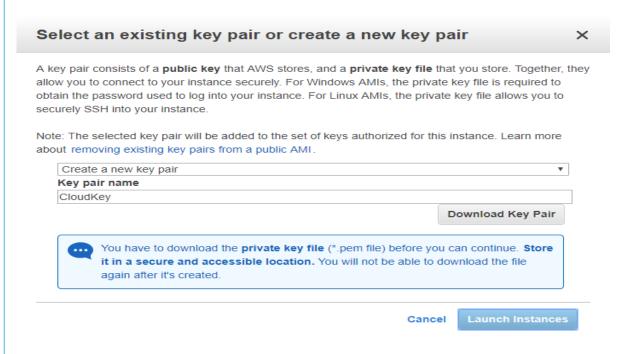
Click Launch





# Key-Pair Creation

- Select Create a new key pair
- Enter key pair name(e.g. CloudKey)
- Click Download Key Pair
- Key (.pem) will be downloaded





## Launch Status

• Click instance id(e.g. i-0535b7...)

#### Launch Status

- Your instances are now launching
  The following instance launches have been initiated: i-0535b7561812a6da9
  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

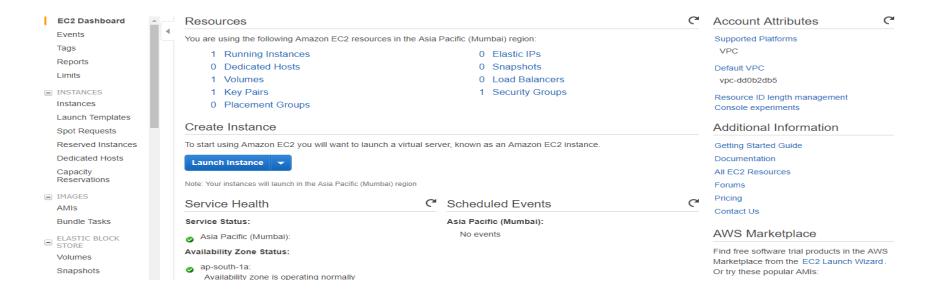
Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

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.



## EC2 Dashboard

Click Running Instances link

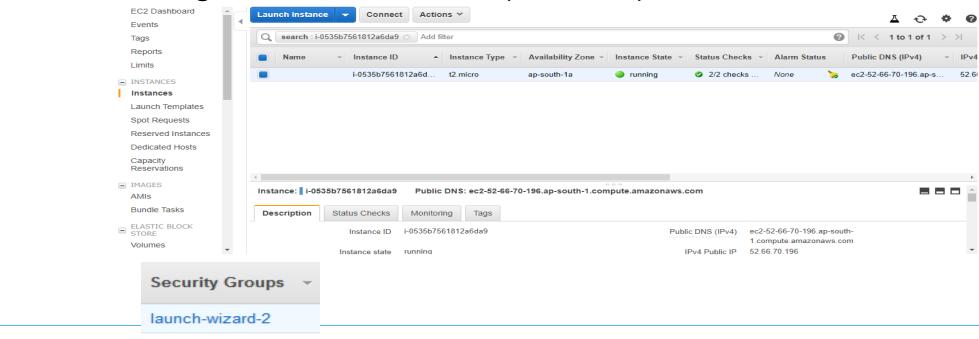




# Running Instance

launch-wizard-3

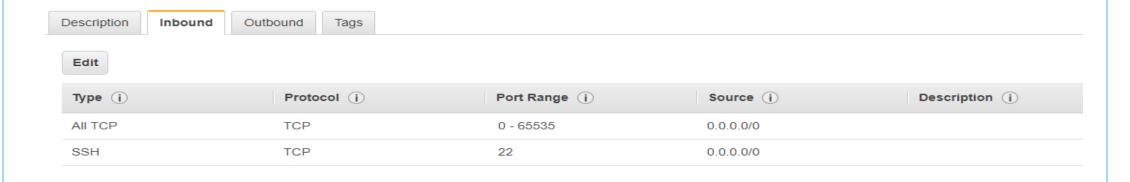
- Drag the page from left to right, you will find Security Groups section
- Click Security Groups link(e.g., launch-wizard-2)
- You will navigate to Inbound section(next slide)





# Create an In-bound rule on AWS

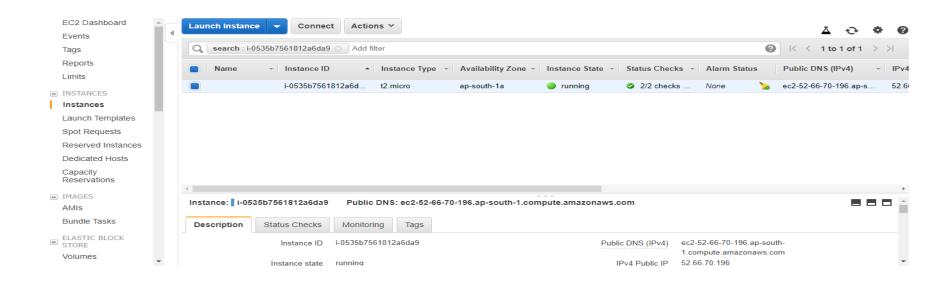
- Click Edit
- Add 2 rules(All TCP)
- Note: SSH rule will be shown in the list default
- Save





# Running Instance

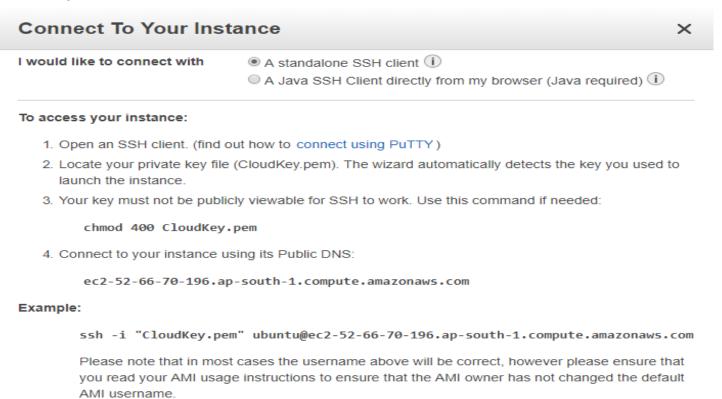
- Come back to running instance page
- Click Connect





# Connecting To Instance

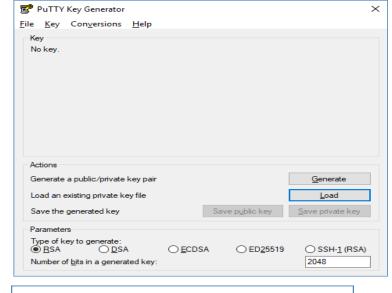
We will use PuTTy to access AWS instance

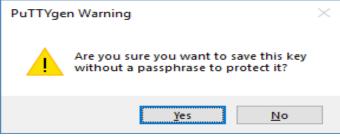


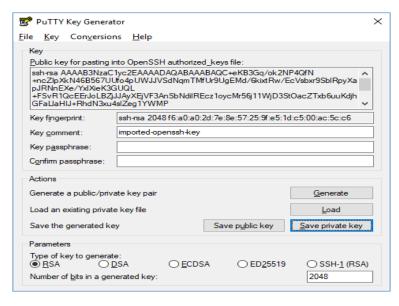


# Using Puttygen to Import Key

- Launch puttygen.exe
- Click Load
- Browse key(.pem) from machine
- Click Save private key
- Click Yes on pop-up
- Browse the location
- (e.g. C:\DO-United\Tools\)
   where you want to save new
   key(.ppk) & Save it
- Close Putty Key Generator window



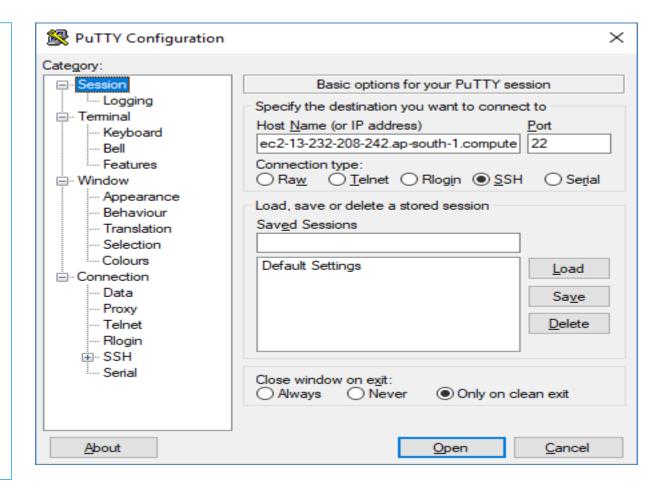






# Using Putty to Connect

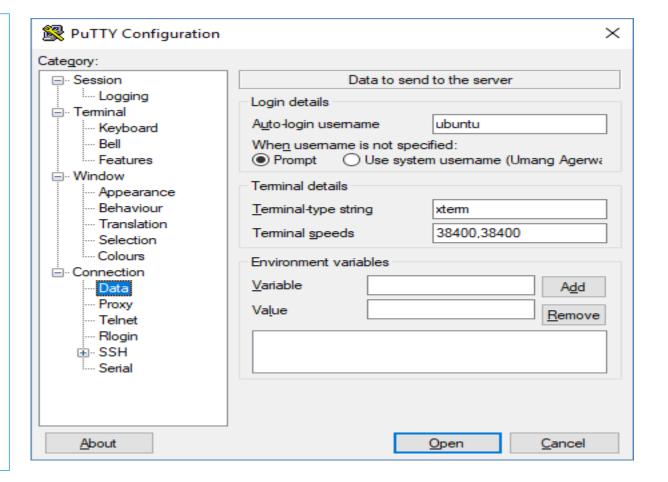
- Launch putty.exe
- Go to Session
- Enter AWS Host Name or IP address(e.g. Ec2-13-233...)
- Note: AWS Host name / IP address will be available on your AWS web account





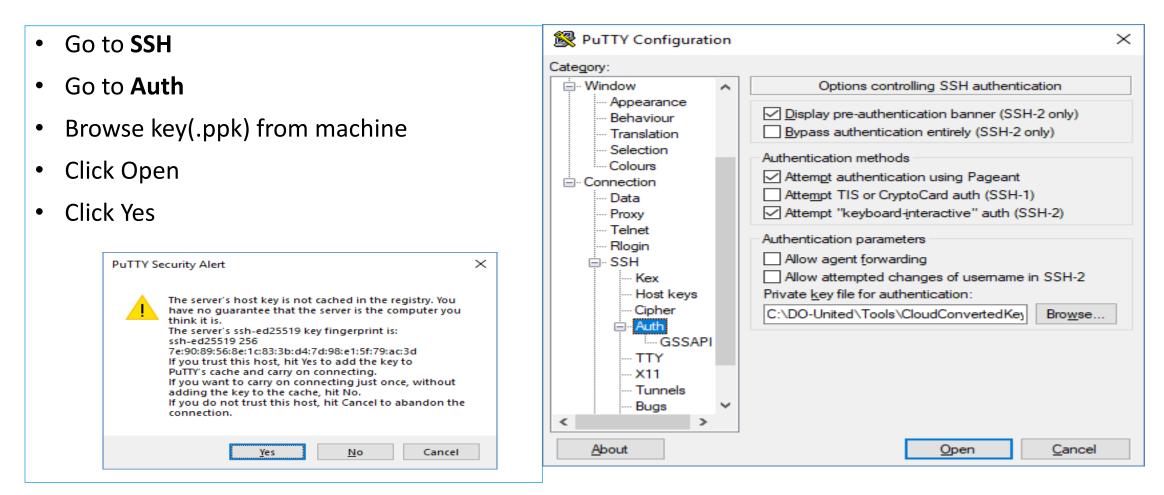
## User → ubuntu

- Go to Connection
- Go to Data
- Enter username ubuntu in Auto-login username section
- Note: For linux machine, enter username
   ec2-user





# Selecting the Private Key





# Connected to Putty

```
ubuntu@ip-172-31-23-141: ~
   - http://bit.ly/Security Certification
 * Want to make a highly secure kiosk, smart display or touchscreen?
  Here's a step-by-step tutorial for a rainy weekend, or a startup.
   - https://bit.ly/secure-kiosk
 Get cloud support with Ubuntu Advantage Cloud Guest:
   http://www.ubuntu.com/business/services/cloud
 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
    https://ubuntu.com/livepatch
63 packages can be updated.
0 updates are security updates.
Last login: Thu Nov 15 08:55:16 2018 from 119.82.80.118
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo root" for details.
ubuntu@ip-172-31-23-141:~$
```





# Setup Docker on AWS Ubuntu machine



## Install Docker on AWS server

- If you create an Ubuntu instance on AWS, run following commands in terminal
  - sudo apt-get update
  - sudo apt-get install docker.io
  - docker --version

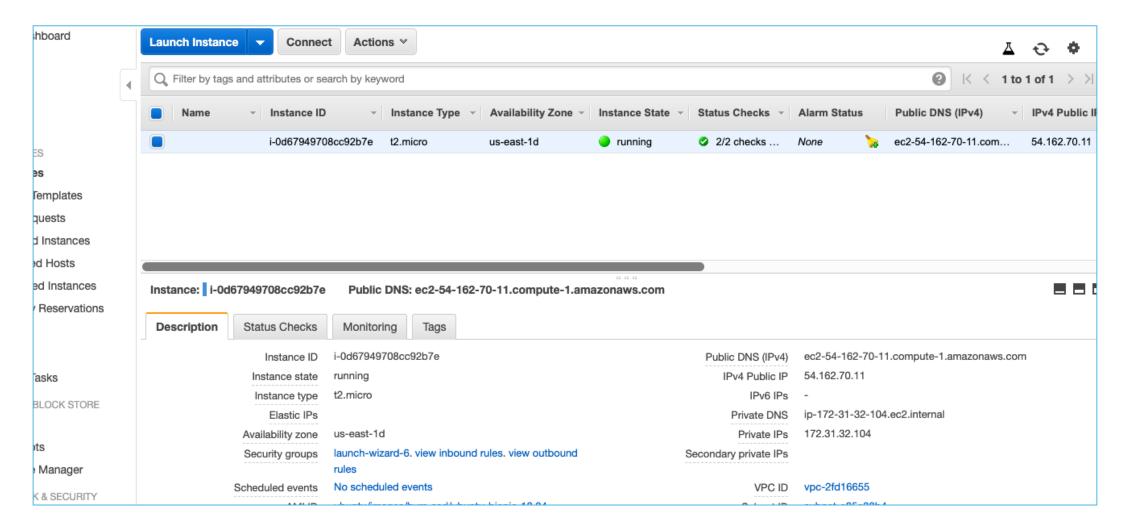


### Run the container on AWS server

 sudo docker run -u root -p 8089:8089 -p 8080:8080 -v jenkinsdata:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock -v \$HOME:/home umangsaltuniv/verity-devops-2-222-3



# Capture the Public DNS/IP Address of the AWS Instance





# Unlocking Jenkis

- After the 2 sets of asterisks appear in the terminal/command prompt window, browse to http://<<IP Address>>:8080 and wait until the Unlock Jenkins page appears
  - This is the same IP which we captured in the previous slide
- From the terminal/command prompt window, copy the automatically-generated alphanumeric password (between the 2 sets of asterisks)
- On Customize Jenkins page click Install suggested plugins
- Click Continue
- Some plugins may be failed so do not worry about that
- IF YOU ARE RUNNING ALL THIS SECOND TIME then
  password will not be available on the screen. See next slide
  for recovering the password if you did not change the
  password from Jenkins UI

**Getting Started** 

#### **Unlock Jenkins**

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

/var/jenkins\_home/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

dministrator password

```
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListab eans [filter,legacy]; root of factory hierarchy Sep 30, 2017 7:18:39 AM jenkins.install.SetupWizard init INFO:

Jenkins initial setup is required. An admin user has been created and a password generated. Please use the following password to proceed to installation:

2f064d3663814887964b682940572567

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
```



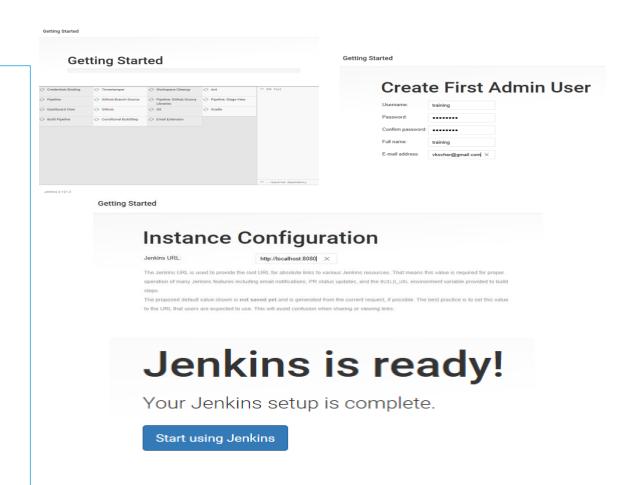
# Getting Jenkins Password

- Launch below commands on terminal:
  - sudo docker container ls
  - sudo docker exec -it <container id> /bin/bash
  - cat /var/jenkins\_home/secrets/initialAdminPassword
- Copy the password



# **Getting Started**

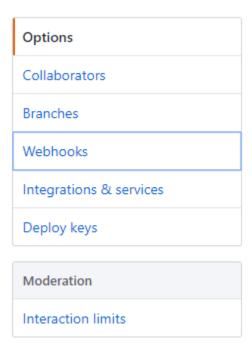
- When the Create First Admin User page appears, specify your details in the respective fields and click Save and Continue. (Here you can change the password)
- Click Save and Finish
- When the Jenkins is ready page appears, click Start using Jenkins OR click Restart, whatever is appeared there.
- Note: If the page doesn't automatically refresh after a minute, use your web browser to refresh the page manually.
- Log in to Jenkins with the credentials of the user you just created





# Setup Webhook on GitHub

- Go to GitHub Web
- Go to "verity-devops-ex" repository
- Navigate to the "Settings" tab
- Select the "Webhooks" option on the left menu
- Click "Add webhook"



Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.



# GitHub Settings

#### For "Payload URL":

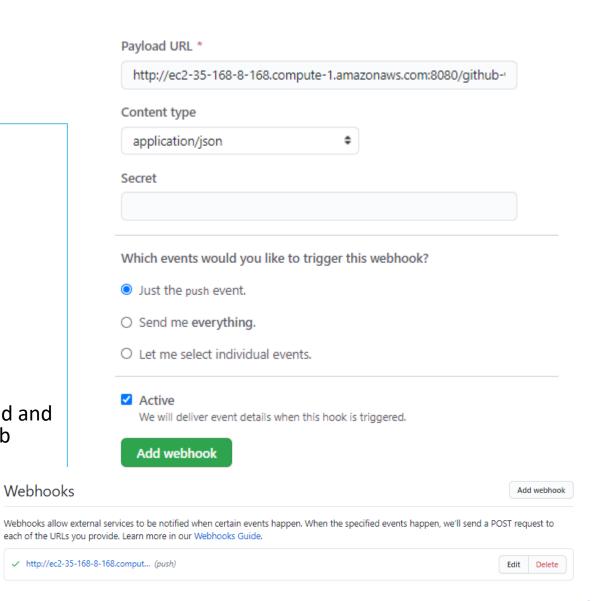
Use the public dns or ip address for the aws machine instance

e.g. http://ec2-35-168-8-168.compute-1.amazonaws.com:8080

• Add /github-webhook/ to the end of it

e.g. http://ec2-35-168-8-168.compute-1.amazonaws.com:8080/github-webhook/

- Select "application/json" as the encoding type
- Leave "Secret" blank (unless a secret has been created and configured in the Jenkins "Configure System -> GitHub plugin" section)
- Select "Just the push event"
- Make sure "Active" is checked
- Click "Add Webhook"





# Create Pipeline Project

- Go to Jenkins Dashboard
- Click "New Item" at top left section
- Enter item name(e.g. verity-devops-ex) > Select Pipeline > Click OK
- Select "GitHub hook trigger for GITScm polling" checkbox under Build Triggers section
- Go to Pipeline section
- Choose the "Pipeline script from SCM" option from the "Definition" field
- Choose the "Git" option from the "SCM" field
- Enter your Repository URL(e.g.)
   https://github.com/<Your username>/verity-devops-ex.git
- Enter "Jenkinsfile.txt" under "Script Path" section
- Click Apply
- Click Save

