

- 1) Click on IAM
- 2) Click on users present in dashboard (left)
- 3) Click on Add users
- 4) Under Username writing 1<sup>st</sup> user pran-iam1 & 2<sup>nd</sup> user pran-iam2
- 5)  Password - AWS Management console access
- 6) Custom password - pran-iam123
- 7) Untick require password reset  (No tick)
- 8) Click Next
- 9) Click [Create Group]
- 10) Group name [pran-iamgroup]
- 11) Under Filter policies give AWScloud9 user rights  
 > AWScloud9 user
- 12) Click [Create groups]
- 13) Click [Next : tags]
- 14) Click [Next : review]
- 15) Click [Create users]
- 16) Copy generated URL & save it somewhere
- 17) Click [Cloud9]
- 18) Click [Create Environment]
- 19) Give name [WebApp IDE demo]  
Description [Let's learn ADEV]
- 20) Click [Next Step]
- 21) Create everything default
- 22) Click [Next : step]
- 23) Click [Create Environment]
- 24) While this environment is creating
- 25) Go to Incognito & login by pran-iam2 by pasting the saved URL

- 26) Go to Cloud9 & wait
- 27) Go to IAM 1 now  $\Rightarrow$  File  $\Rightarrow$  new from template  
 $\Rightarrow$  HTML file  $\Rightarrow$  write title (anything)  
 Filename: first.html
- 28) Save as first.html  $\Rightarrow$  save
- 29) Click share
- 30) Invite members
 

iam-1	IAM 2	Give [rw]
-------	-------	-----------
- 31) Invite (click)
- 32) Security warning OK (click)
- 33) Click Done
- 34) Now go to IAM 2
- 35) Click on 3 dots Shared with you
- 36) Click Open IDE
- 37) Open first.html
- 38) Now show both side by side
- 39) Both will work & edit & at real time
- 40) Take screenshot (can also show group chat)

## Exp-2 AWS codePipeline

### Part I

- ① Go to Elastic Beanstalk
- ② Click [Create Application]
- ③ App name [pran Demo Web Env]
- ④ choose a platform [Tomcat] (Create all will be auto filled)
- ⑤ Sample Application
- ⑥ Click [Create Application]

⑦ Environment will be created (will take some time)  
then below given & can be seen

pran Demo Web Env

URI	Health	running version	platform
	OK		Tomcat [change]

### Part II of Exp

### ⑧ Github

↳ Go on man's repository link & Fork it

### Part III

- ⑨ Come to AWS
- ⑩ Click on code pipeline
- ⑪ Click [Create Pipeline]
- ⑫ Pipeline name [pranDemoCodePipeline]
- ⑬ Every other thing default
- ⑭ Click [Next]
- ⑮ Source provider [GitHub (version 2)]
- ⑯ Click [Connect to GitHub]

→ [instead new app]

(17) Enter connection name

[Github email] → [connect]

(18) [Authorize] click

(19) ⚡ Only select repository

[Select required] repository

(20) [Install] click

(21) [connect] click

(22) Repository Name will be filled auto

(23) Branch name main

(24) [Next] click

(25) Build provider AWS codebuild

(26) Create Project click

(27) Project name pranCodeBuild

(28) OS Ubuntu

(29) Runtime Standard

(30) Image aws/codebuild/standard:4.0 (Pre-built)

(31) Build spec ⚡ use a buildspec file

(32) Rest all default → [Continue to CodePipeline] click

(33) Build type ⚡ Single build

(34) ⇒ [Next]

(35) Deploy provider AWS Elastic Beanstalk

(36) Application name Indu Demo WebEnv

(37) Environment name Auto filled

(38) ⇒ [Next]

(39) Review everything → [Create Pipeline]

(40) Under



there will be link as



Deploy AWS Elastic Beanstalk click

PAGE NO.	
DATE	/ /

- (41) click on URL (http://) & click
- (42) Index.html file will get displayed
- (43) Now go to ~~to~~ GitHub & edit index.html  
& commit it
- (43) source  $\Rightarrow$  Build  $\Rightarrow$  Deploy (will happen)

## Exp3 - Kubernetes Installation

- 1) Go to EC2
- 2) Click on Launch Instance (Opt into old experience)
- 3) Number of instances (under summary) [2]
- 4) Select Ubuntu Server 20.04 LTS (Free tier)  
Step 2!
- 5) Choose instance type  
t2 micro (Free tier)
- 6) Next : Configure instance details click
- 7) Step 3 : Configure instance details  
Number of instances [2]
- 8) Next : Add storage click
- 9) Next : Add tags click
- 10) Next : configure security group  
click [Add rule]
 

Type	-> Port no 6443-	Source
[HTTP], [HTTPS], [Custom TCP]		[Anywhere], [Anywhere], [Anywhere]
- 11) Review & Launch
- 12) Launch
- 13) Create a new key pair
  - ↳ RSA
  - ↳ Key pair name [kubernetes]
  - ↳ Download key pair
- 14) Launch instance
- 15) Now click on instances
  - ↳ Name 1 of them as "K8S Master"
  - ↳ Name 2<sup>nd</sup> as "K8S Worker"
- 16) Click on [Connect]
  - ↳ Click of  K8S master & click on [Connect]
  - ↳ Copy Public IPV4 DNS
  - ↳ skip below
- 17) Open MobaXterm => User Session => New Session => Put Copied IPV4 DNS under Remote host

Master  $\Rightarrow$  green  
worker  $\Rightarrow$  red

PAGE No.	
DATE	/ /

(18)  Specify username  port [22]

(19) Now under Advanced SSH settings

USE private key

(20) Bookmark settings  $\Rightarrow$  session name   
 Customize tab color  green

(21)  Click

(22) Do same process with K8S worker & with same downloaded key but session name  & color  red

(23) Under K8S master & worker in MobaXterm  
write clear & sudo su  
 $\Rightarrow$  Above is for MobaXterm

After (6)

(17) After clicking on [connect]

(18) ~~For instance connect window will pop up~~  
do nothing  $\Rightarrow$  click [connect]

(19) Same step of selecting  k8s worker

(20)  $\Rightarrow$  click [connect]  $\Rightarrow$  Connect to Instance Pg =  
Again click [Connect]

Under master cmd

(21)  $\hookrightarrow$  write clear

$\hookrightarrow$  write sudo su

hostnamectl

hostnamectl

ubuntu # hostname1 set-hostname master-node  
-+ # exit

hostname1

-+ # sudo su

-+ # clear

after /- (space) in Software Repository

PAGE No.	/ /
DATE	

(22)

under worker

```
↳ /ubuntu# hostnamectl set-hostname worker0  
→ → # exit  
→ → # sudo su  
→ → # clear
```

(23)

master color change

```
↳ # tput setaf 02
```

(24)

→ worker color change

```
↳ # tput setaf 01
```

(25)

master / worker (both)

```
↳ # apt-get update  
↳ # apt-get install docker.io → -y  
↳ # docker -version
```

(26)

Enable & Start Docker (both)

```
↳ # systemctl enable docker
```

```
↳ # systemctl status docker (press Q)
```

```
↳ # systemctl start docker
```

(27)

Do all other steps till [step 7]

(28)

Step 8: on Master only

(29)

Perform all other commands as given in document

## Exp 4 - Kubernetes cluster & Deploy

PAGE No.	/ /
DATE	

- 1) create deployment named nginx nginx on master
- ↳ # kubectl create deployment nginx --image=nginx
  - ↳ # kubectl expose deploy nginx --port 80 --target-port 80 --type NodePort
  - ↳ # kubectl get services

2) Here we will get port no under Port column, which is greater than 30000

name	type	---	Ports	age
kubernetes				
Nginx			80:nginx port eq 36276	

3) Now goto AWS => EC2 => Instances

K8S worker (Select)

under details

Public IPv4 address

54.242.161.135 | open address (It's link, click if)

4) Now add port no to that link address,  
eq 54.242.161.135 : 32670

5) It will not work becoz we did not give access

6) Under instance =>  K8S worker => Security => Click on security group (link => Inbound rules) => [Edit Inbound rules] => [Add rule] > give port no 32670 Anywhere => [Save]

④ Now it will run & show (welcome to nginx) # note use http & not https on http:// URL : port no

⑤ scale up replicas (to make multiple web servers)  
(write commands given in document  
see & under Master)

- \* Go to For demo  
 If we are not getting output then we can show demo)
- ① Go to Nagiosxi.demos.nagios.com
    - ↳ Show Service Status Summary (SS)
    - ↳ Show critical => Showdown (SS)
    - ↳ Show my reports (SS)
    - ↳ Show Views
    - ↳ Show Network Status Map
    - ↳ Show configuration Wizard
  - ② Go to official nagios website & download
    - ↳ Nagios XI (.ova file)
    - ↳ VM ware (v16 or v17 latest) (for windows)
  - ③ Open VM workstation
  - ④ Click on Open a virtual machine
  - ⑤ Select 'the downloaded .ova file'
  - ⑥ Name [Nagios]
  - ⑦ Choose drive which has space [Browse]
  - ⑧ [Import] click
  - ⑨ Now Nagios installation is done
  - ⑩ Click on Nagios below Home
  - ⑪ ▷ Play virtual Machine (click)
  - ⑫ Now the IP below is to be copied & paste on URL
  - ⑬ Note: If IP is not visible
    - Right click on nagios in vmware startup and check setting in hardware, go to network adapter.
    - in network connection, change it to NAT Used to share the host IP address
    - Click ~~OK~~ OK and start your VM, ip addr should appear

(14)

Now login

↳ Select Timezone [mumbai]

↳ Language [Eng]

↳ User Interface Theme [Modern]

↳ Licence type • Free

Change

(15)

Now give new login / password

(16)

Now login with it

(17)

Access of Home dashboard (Installation done)

(18)

Now in Host Status Summary → Click up

(19)

Show Network

### part B : Monitor Remote Host / PC using NRPE

(20)

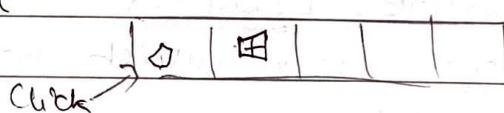
Now in Nagios XI

(21)

Go to configure

(22)

Select Linux



Click

(23)

Select Linux server (Legacy) (One with NRPE)

(24)

IP address

(25)

Linux Distribution

Ubuntu

(26)

Open Ubuntu (by VMware)

Now Ubuntu system is to be monitored

↳ Go in terminal of Ubuntu

↳ Write ipconfig

Now copy the IP given → inet 192.168.29.13  
name eg

(27)

Paste it in (23) IP address

PAGE No.	
DATE	/ /

- (28) Now click [next]
- (29) Give Host name [nagiosclient]
- (30) Now perform give commands by man  
given in Nagios XI - Installing the Linux agent  
Search in chrome