AKASH B. KOCHURE

B.E. (Mechanical Engineering)

Objective

Want to work in an organization where my skills and hard work will lead the organization to next level. And I can learn new things from that organization which will lead me towards better and better future.

College Project Details

Design & Fabrication Of Rust Removal Vibrating Tumbler Machine:

Background and Motivation: We made a pot shape machine in which finishing is carry out by using vibrating motion. The abrasive particles are collide with the rusted material and remove the rust by vibrating motion.

By using this method high degree of finishing is possible with less stock removal. Polishing of small parts can be done. It is very useful for hidden rust such as thread in nut & bolt.

Education:

J.T. Mahajan College Of Engineering, Faizpur

- BE in Mechanical Engineering from NMU in 2017.
- Secured aggregate 6.95 CGPA = 62.00%.

Indirabai lalwani College, Jamner

- HSC(2012) in Science Stream From Nashik Board.
- Secured 60.50%.

Smt.S.P.Rane High School, Khirdi Kh.

- SSC Examination (2010).
- Secured 88.91%.

Certificates:

AWS Solution Architect-Associate & DevOps corporate training completed from 3RI Technologies.

30WPM Typing, DTP & MS-CIT.

Mindset

Minimalist.

Good Listener, Good Communication.

Positive Nature, Team Player, Team Leader.

Time Management, Target Orientantion.

Hard Work, Sincerity is my key element.

Personal Information:

Mobile No.:

9561109276

E-mail id:

akashb2694kochure

@gmail.com

Date of birth

06/02/1994

Skills

Oracle Vm-Box Setup

SQL & Databases

Linux/Unix & Linux Commands

Windows, Debian, Mac-Os, Redhat etc.

Ubuntu & Shell Scripting

Network-Architecture Diagrams & Planning

DevOps (Git, Jenkins, Kubernetes, Docker)

DevOps Projects Deployment & Management

AWS Services & Each Modules of AWS

AWS Project Management & Monitoring

Interests

- **Cloud Computing.**
- **Networking & Domain.**
- **Azure & GCP & Alibaba.**
- **IBM & Apple & Oracle.**
- **Vm-Ware & Sales-force.**
- **AI & APIs & Migration.**
- **Coding & Programming.**

Languages

- English
- Marathi
- Hindi

TECHNICAL – SKILLS

- Worked on Amazon Web Services (EC2, ELB, VPC, S3, **CloudFront**, IAM, **RDS**, **Route 53**, CloudWatch, **SNS, SQS, Lambda**).
- Installed and Setup Web Servers (Apache and **Tomcat**), DB Server (MySQL).
- Design and deploying cloud infrastructure as per SOW & Create **Network-Architecture-Diagrams** Using **Draw.io** tool.
- Launch and managing Amazon **EC2** and **VPC** instance.
- Creating load balancer (**ELB :-** ALB, NLB & CLB).
- Creating and managing files and polices in **S3 bucket** (S3-Versioning, Cross-region Replication, Static Website Hosting & **CORS**).
- Creating and managing VPC, Subnet and Security groups, **ACLs** & **NACLs** & **Firewalls**.
- Create Users, Groups, Provide IAM Roles and Policies to the user & group. Providing **MFA** Verification to the User.
- Create and providing **IAM access** to the user. Implementing **Auto Scaling-in** & **Scaling-Out**.
- Creating EC2 Instances using **AMIs & SnapShots**.
- Set **Billing-Alarms** & **Custom-Budget-Alarms** and Create **Aws Cost & Usage Reports**.

- **Cloud Infrastructure:**
  Amazon Cloud: EC2,**S3**, **EBS**, **EFS**, RDS, DynomoDB, VPC, **VPN**, **Quick-Sight**, **EMR**, **ETL**, **Cloud Formation, Elastic Beanstalk**, AWS **CLI** and AWS **SDK** & AWS **CDK** for Python and Ruby, **Snapshot**, Managing **Back-Up Policy**, Redshift, Redis, **Athena**, **AWS Glue**, AWS Crawler, **API Gateways, Light-Sail, OpenVpn, Cloud9, ECS, EKS, AWS KMS, CloudWatch, CloudTrail, DNS** etc.

- **DataBase:** PostgresSQL and MySQL.

Storage: Amazon S3 and **Glacier**.

- Proficient with Miscellaneous **Git**, **GitHub**, **Bit-Bucket**, **Sub-Git** & **SVN** & **CVS** & **Mercurial** for versioning management system.
- Experience with server-side technologies such as **Apache**, **Nginx**, **Apache2**, **Apache-Tomcat** for java web pages & html or image / contains.
- **Maven** is used to build Java based configuration Project by using POM.xml files, VCS and running builds using **Jenkins** & **Bamboo**.
- Knowledge of software containerization platform like **Docker** and **container orchestration** tools like **Kubernetes** & **docker-swarm**.
- Proficient in developing **CI / CD (Continues Integration/Delivery) pipeline**.
- Good Understanding of Infrastructure as code(Programmable Infrastructure), and how you can achieve that by using **Ansible** and **Puppet**.
- Build and Deployed **Docker containers** for implementing Microservice Architecture from Monolithic Architecture.
- Experience with server monitoring systems like **Nagios**, **Splunk**, **Prometheus**, **Grafana** & **ELK Stacks**(Elasticsearch, Logstash, Kibana) etc.
- Server automation with **Terraform** and **Ansible** & **Puppet**.
- Database (Mysql primarily) administration experience.
- Ability to work within a Team with strong analytical, problem solving and communication skill.
- **SonarQube** for code analysis & **Jfrog / Artifactory** for build, test, release, deploy and continuous Improvement using APIs.
- Able to work in ambiguous situations. Flexible and adaptable.
- **Helm / Helm-Chart** for deploying and managing Kubernetes apps.
- Core skills for **SDLC** with DevOps **&** Waterfall **&** Agile, Scrum-Master**,** Kanban technologies using Jira in **SRE** as Project Management tool.
- Basic knowledge of other tools like **Vagrant**,**Nexus**,**Salt-stack**,**Splunk**,**Graddle**,**Travis**,**TeamCity**,**Selenium**,**TestNG**,**JUnit**,**New-Relic** etc.
- Proven leadership, management and presentation skills in Infrastructure service domain and client-facing engagements.
- Streamlined deployment process by developing continuous integration tools like Jenkins.
- Maintained Git repositories for DevOps Environment. Version control and build automation integrating git into Jenkins.
- Reduced time to deploy applications using DevOps concepts like configuration management and continuous deployment using Ansible.
- I can works in Production support 24*7, Web Application, Cluster building and configurations.
- Responsible for building and deploying applications in build, test and non-production environments.
- Experience in handling high severity issue. Performed regular backups for deployed applications.
- Deployed applications such as **.ear and .war** files and tested them in QA and production environmental.
- Assembled and Deployed the application in staging, in productions, following change management practices and 24/7 support, worked in shifts.

● Skills performing operational tasks and activities (e.g. Monitoring log files, performance tuning, application monitoring) managing and monitoring the performance and troubleshooting issue.

● Backup and Recovery implementation.

● Skills in providing round the clock on call support for production issues.

● Excellent skills in maintenance and production support related projects.

AWS Labs - Getting Hands-On

Lab-1) AWS SITE-TO-SITE VPN Configuration & Create Tunnel for Private Ip.

Lab-2) AWS **Client-VPN Endpoint** Configuration Lab.

Lab-3) AWS **VPC-End-Point** Configuration For **S3 Bucket** / DynamoDB / Athena Lab.

Lab-4) **Nat-Instance** Tutorial Lab **& EFS / NFS** File System Tutorial Lab.

Lab-5) **VPC Direct-Connect** Lab.

Lab-6) Hello-World labs on Linux & Windows Instances. Install & Run Apache, Apache2 & Tomcat web-server.

Lab-7) **VPC-Peering** between same/different region labs on Linux & Windows Instances.

Lab-8) VPC-Transitive-Peering between same/different region labs on Linux & Windows Instances.

Lab-9) **REDSHIFT :-** Deploying a Data-Ware-House with Redshift.

Lab-10) ELASTIC-BEANSTALK :- Create a Infrastructure with Elastic-Beanstalk with Python Code / JAVA + S3 Bucket + Load Balancer + Cloud Environment.

Lab-11) LIGHTSAIL :- Hosting a Hello-World / Word-Press Web-Site on AWS LightSail.

Lab-12) SNS + SQS + LAMBDA :- Aws SQS + Lambda Function & send SMS Tutorial Lab.

Lab-13) CLOUD FORMATION :- Tutorial Lab.

Lab-14) **AWS CLOUD-9 (AWS-Code-Commit) :-** Tutorial Lab.

Lab-15) OPENVPN :- OpenVpn Set-Up & Create Multiple Client user hosting static Ip.

Lab-16) REST-API-Gateway + AWS Lambda Function :- Create Hello-World from Lambda / Student Record / Crud API With Lambda and Dynamodb Lab.

Lab-17) Word-Press + AWS RDS :- Deploying WordPress site over AWS using RDS Lab.

Lab-18) XAMPP / WAMP / LAMP / MAMP :- XAMPP / WAMP / LAMP / MAMP Web-Server Installation in AWS EC2 Instances Lab.

Lab-19) Purchasing Domain from GoDady Website and Transfer Domain on AWS Route53 & Create Hosted-Zone.

Lab-20) Performing Various labs using **DevOps Tools** Like **Terraform** / **Ansible** / **Docker** / **Jenkins** / **Maven** / **Nagios** / **CI / CD Pipe-Line** / **Git / Git-Hub** / **Kubernetes** with **Visual-Studio-Code** & Creates Containers & Infrastructure.

SET – UPS & PROFICIENCY

1) Proficient in **Oracle VIRTUAL-BOX** (Vm-Box) Setting with Ubuntu (32/64 bits).

2) **SQL DEVELOPER** with Oracle.

3) **SQL/Plus *** With Oracle.

4) **PL/SQL** Set-up with Oracle.

5) **MYSQL-Workbench 8.0** + **Shell 8.0** + **Server 8.0** Set-up on Windows 11.

6) Connecting **Amazon RDS** with **SqlElectron** as well as with **EC2 Instances**.

7) **PYTHON** Setup.

8) API – POSTMAN App Installation & Set-up.

9) Proficient in creating Infrastructure like EC2 instances and S3 Bucket by using **Terraform**.

10) Proficient in **Docker, Docker-Hub Repository, Docker Container , Docker-Compose & Docker-Swarm**.

11) Proficient in **Kubernetes, Kubernetes Orchestration, Minikube, Kubernetes-Cluster, Kubernetes master & Slave, Nodes & Pods** etc.

12) Proficient in **Linux** & **Linux Commands**. Proficient in **Ubuntu** & **Shell-Scripting**.

13) Proficient in **Ansible**, **Ansible Playbook**, **Ansible-Master & Salve** etc.

14) Proficient in **Sql** & **Databases** (Amazon RDS, Mysql, Athena, Mariadb, Postgres, Redshift, Redies, DynamoDB, MongoDB, DocumentDB, Hadoop, Apache-Spark).

15) Proficient in **CI/CD PipeLine, POM.xml, Jenkins, Git , GitHub , Maven, Chef , SonarQube, Puppet, Nagios , Prometheus & Grafana and Tomcat Servers**.

16) Proficient in Handling, Developing , **Managing** & **Monitoring DevOps Projects**.

AWS - PROJECTS - DETAILS

Project 01 : CREATING A FILE SHARE & SYNC SOLUTION USING OWNCLOUD AND AWS.

Description :- 1) Create Owncloud-VPC, Create Public & Private Subnet.
2) In Public EC2 instance Install Apache2, Php & Owncloud App.
3) In Private EC2 Instance Install MysqlDB.

Project 02 : 3-TIER-ARCHITECTURE – PROJECT 01 – WEBSERVER / WEBHOSTING PAGE.

Description :- 1) Create VPC, Create Public Subnet, Create Private-01 & Private-02 Subnet.
2) In Public Subnet Install Webserver / Webhosting Page.
3) In Private-01 Subnet Install Apache2 & Php.
4) In Private-02 Subnet Create **Amazon RDS – MySqlDB**.
5) Appy **Autoscaling-Group** (Minimum-02).

Project 03 : 3-TIER-ARCHITECTURE – PROJECT 02 – XAMPP-SERVER / PHPMYADMIN.

Description :- 1) Create PhpMyadmin-VPC.
2) Create Public Subnet – (MyWeb-01 , MyWeb-02 , MyWeb-03) :- Install PhpMyadmin.
3) Create Private Subnet – (MyApp-01 , MyApp-02 , MyApp-03) :- Install Apache2 & Php.
4) Create Private Subnet – (Mysqldb-01, Mysqldb-02, Mysqldb-03) :- Install Mysqldb.
5) Apply **ALB** (**Application Load Balancer).**

Project 04 : BIG – DATA – FABRIC / (AWS GLUE + ATHENA + QUICKSIGHT) PROJECT.

Description :- 1) Create S3 Bucket, Create Folder & **Upload .cvs File.**
2) **AWS Glue + Crawler**.
3) **Athena** (Query Reg. & Manage & Query Editor).
4) **Quick-Sight.** (Quick-Sight Reg. & Use).

Project 05 : SERVER-LESS / MULTI-TIER / N-TIER ARCHITECTURE .

Description :- 1) **AWS Lambda**.
2) **API Gateway**.
3) **DynamoDB**.
4) Java.
5) **Route53**.

DevOps – PROJECTS – DETAILS

Project 01 : Complete “Hello-World” CI/CD Project using Git, Maven, Jenkins-Server and Tomcat-Server.

Scenario :- Setup : here we will use GIT+Jenkins+Maven+Tomcat together and configure it as CI-CD flow:

Login to Jenkins console
Create Jenkins job, Fill the following details,

source code management::
Mention the repo URL and the master details in it.

IN BUILD: select Maven version (invoke top level maven targets)
Root POM: pom.xml
Goals : clean install package

Adding Deployment Steps:
in this case we are going to install 'deploy to container' plugin. this is need to deploy on tomcat server which we are using.

Install maven plugin without restart
Manage Jenkins > Jenkins Plugins > available > deploy to container

on tomcat server our Jenkins server need access. For this we should setup credentials. This option is available in Jenkins home page:

setup credentials

credentials > jenkins > Global credentials > add credentials
Username   : deployer
Password : XXXXXXX
id : Tomcat_user

Description: Tomcat user to deploy on tomcat server

Modify the same job which created:
Post Steps
Deploy war/ear to container
WAR/EAR files : **/*.war   --------- in workspace dir wherever it finds war file ext..it will take that
Containers : Tomcat 8.x
Credentials: Tomcat_user (which created in above step)
Tomcat URL : http://<PUBLIC_IP>:<PORT_NO>/webapp

Save and run the job now.

Continuous Integration & Continuous Deployment (CI/CD)
Now job is running fine but to make this as Continuous Integration and Continuous Deployment we have to add below steps:

Build Triggers
Poll SCM
schedule */2 * * * *
Save the job and modify the code in GitHub.

Save the job and modify the code in GitHub. Then you could see your job get trigger a build without any manual intervention.

Project 02 : Complete CI/CD Project using Ansible, Git, Jenkins, Maven & Tomcat-Server.

Scenario :- Setup : ansible+git+EC2+Maven

Install "publish Over SSH" plugin and restart jenkins if required: (Manage Jenkins > Manage Plugins > Available > Publish over SSH)

Enable connection between Ansible and Jenkins:

for best practice:
create a generic user on both the ansible instance and tomcat instance:

```
useradd -m -d /home/ansibleuser ansibleuser
```

set password for that user:

```
passwd ansibleuser
```

Manage Jenkins > Configure System > Publish Over SSH > SSH Servers
SSH Servers:
Hostname:<ServerIP>
username: XXXXXXX
password: *******

Test the connection "Test Connection"

login to ansible VM:
create a folder in /opt/playbooks once logged in via "ansibleuser"

create a yml file which will copy the stuffs to several tomcat instances:

```
---
 - hosts: tomcat
   become: true
   tasks:
         - name: copy war file onto tomcat instance
           copy:
                 src: /opt/playbooks/target/webapp.war
                 dest: /opt/apache-tomcat-8.5.40/webapps
```

PS: add the internal IP's of the tomat instances inside /etc./ansible/hosts and make sure that few ad-hoc commands get executed:

```
ansible -m ping all
```

Jenkins Job create:
we can craete new job adn use settings to copy from the latter job:

goto : post steps:
send files or publish over ssh (this will come if plugin was installed)

transfer:
source file: webapp/target/*.war
remote dir : //opt//playbooks

add 1 more publish over ssh:

exec command we need to add and leave other as blank:

```
ansible-playbook /opt/playbook/copy.yml
```

Add post build steps:

Execute job and you should be able to seen build has been deployed on Tomcat server.
try accessing the page via :
public IP of tomcatinstance:8090/webapp

Project 03 :- Complete CI/CD Project using Docker, Git, Jenkins, Maven & Tomcat-Server.

Scenario :- Setup : Use the same feature for Docker instead of EC2 instances :-

Prerequisite::
an EC2 instance for Docker host (ubuntu preferred)

Docker installations:

```
yum install docker
service docker start
```

or:

```
apt install apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
apt update
apt-cache policy docker-ce
apt install docker-ce
```

new user for Docker management and add him to Docker (default) group::

```
useradd dockeradmin
passwd dockeradmin
usermod -aG docker dockeradmin
```

Please make sure that in Jenkins VM,the ssh is allowed via using password..

login to docker host now and create a docker file now:

```
From tomcat:8-jre8

# copy war file on to container
COPY ./target/webapp.war /usr/local/tomcat/webapps
```

Login to Jenkins console and add Docker server to execute commands from Jenkins:
Manage Jenkins --> Configure system --> Publish over SSH --> add Docker server and credentials

Create Jenkins job:
after adding all git details
send files or execute commands over SSH Name: docker_host

source files: target/*.war
remove prefix : target (this will remove the target name while editing the war file)
remote directory : //opt//docker
exec command: docker stop jenkins_docker;docker rm -f jenkins_docker;docker image rm -f jenkins_docker;cd /opt/docker;docker build -t jenkins_docker .

send files or execute commands over SSH (its always advised to add 2 times as commands in single can fail)
Name: docker_host
exec command : docker run -d --name jenkins_docker -p 8090:8080 jenkins_docker
save the job

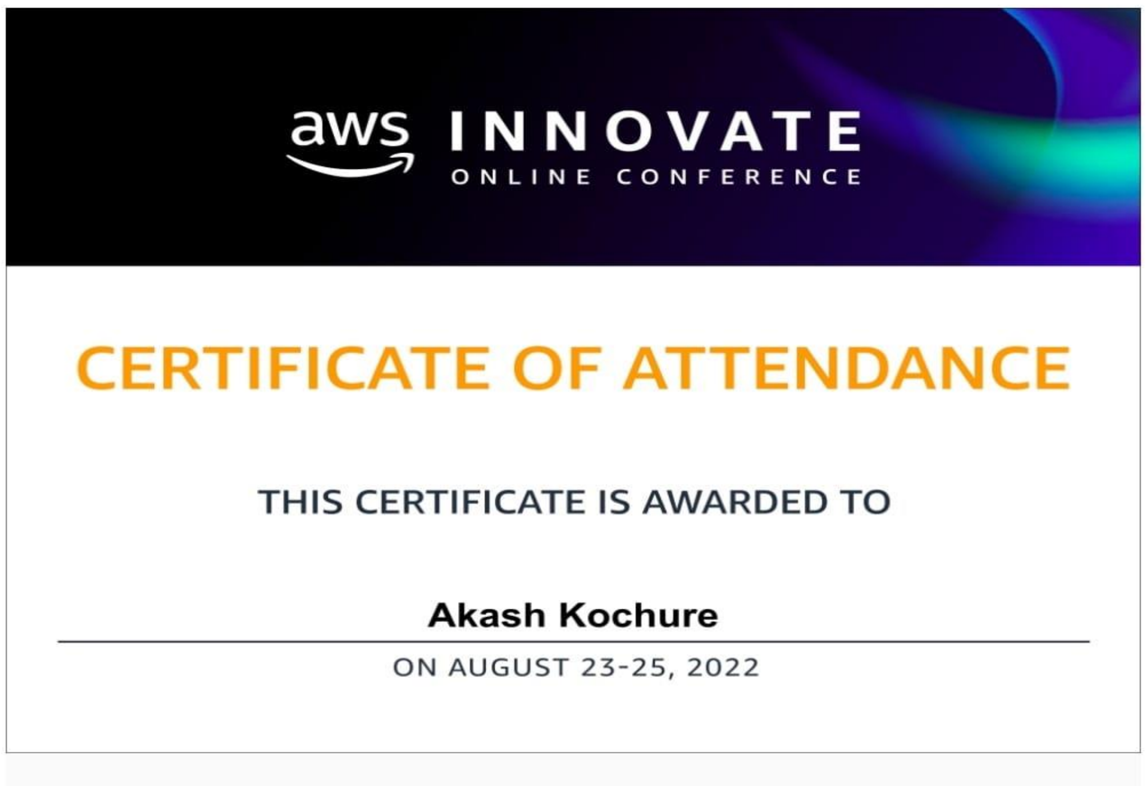
try accessing if tomcat page gets opened:
public IP:8090
later for the application : pubIP:8090/webapp

Declaration:-

I hereby declare that above information is true and correct to the best of my knowledge and belief. Reference could be provided on request.

Place: Pune.

Date: **(Name / Signature)**



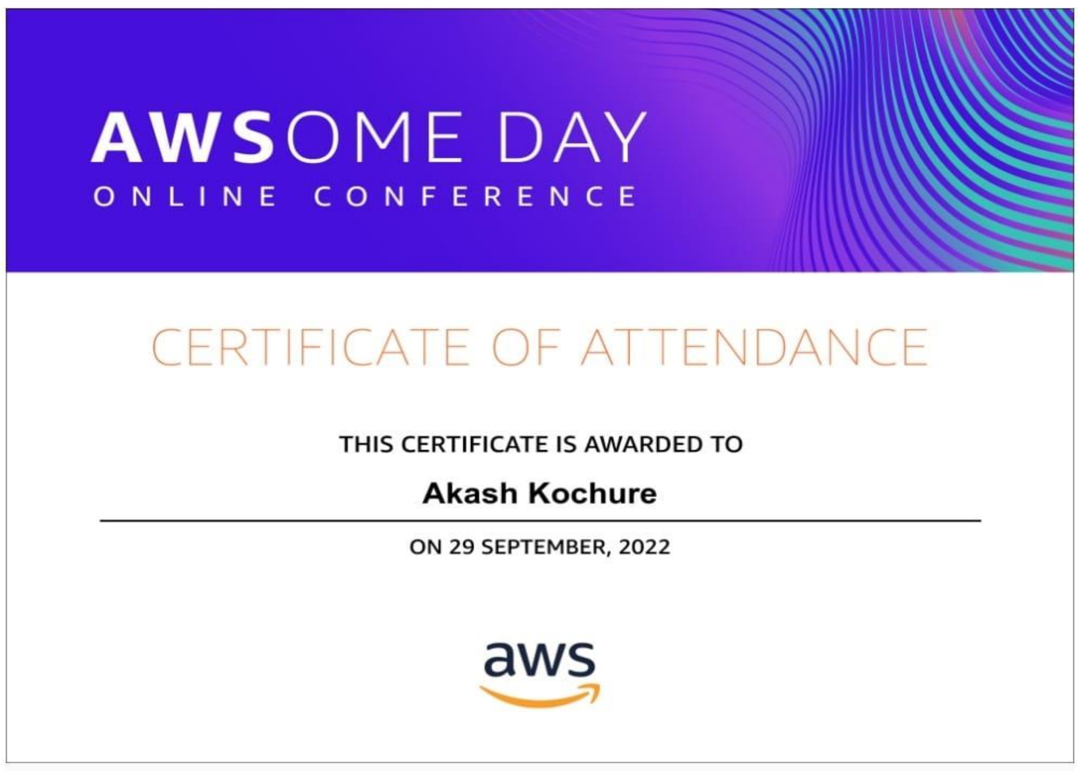
aws INNOVATE
ONLINE CONFERENCE

CERTIFICATE OF ATTENDANCE

THIS CERTIFICATE IS AWARDED TO

Akash Kochure

ON AUGUST 23-25, 2022



AWSOME DAY
ONLINE CONFERENCE

CERTIFICATE OF ATTENDANCE

THIS CERTIFICATE IS AWARDED TO

Akash Kochure

ON 29 SEPTEMBER, 2022

aws