

DevOps

Robot Shop: A Comprehensive Exploration of its 3-Tier Architecture, 8 Services, and 2 Databases



mrcloudbook.com · 30 December 2023



Mr Cloud Book

[Home](#) [Blog](#) [DevSecOps](#) [Contact](#) [About Me](#) [Testimonials](#)

[Search Blogs](#)

Today, let's embark on a journey of deploying Stan's Robot Shop, an educational microservices application. This sandbox environment serves as an excellent playground to delve into the realm of containerized applications, exploring deployment methodologies in a practical manner.

COMPLETE CREDITS ABHISHEK VEERAMALLA

<https://youtu.be/8T0UnSgywzY?si=cZDDK09klAd-2vul>

Contents [hide]

STEP1: CREATE IAM USER IN AWS

STEP2: Create EC2 Instance

Step3: Connect to Instance and Install Required Packages

STEP4: EKS Setup

Commands to configure IAM OIDC provider

Check if there is an IAM OIDC provider configured already

Setup alb add on

Deploy ALB controller

EBS CSI Plugin configuration

STEP5: DELETE CLUSTER

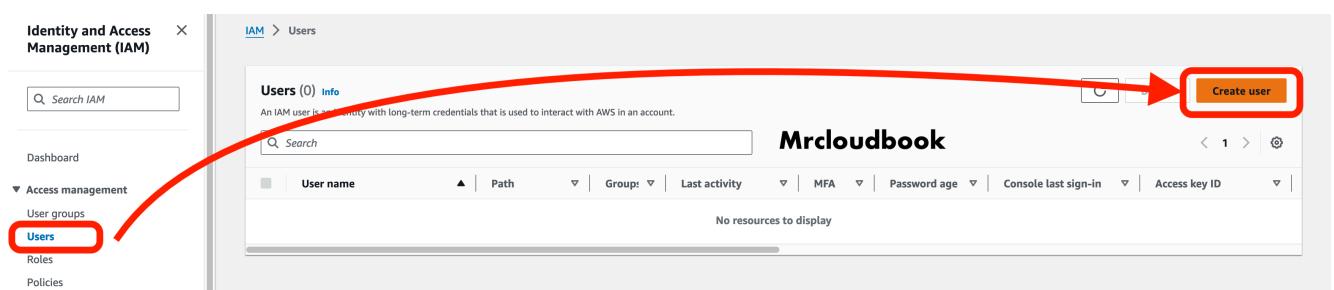
STEP1: CREATE IAM USER IN AWS

Go to Aws console and login with your credentials

IN Search bar TYPE IAM

This is IAM Dashboard

Click on Users and click on Create User



Provide a Name and provide checkboxes and click on Next

User name
 3tier
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Provide user access to the AWS Management Console - *optional*
 If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type

- Specify a user in Identity Center - Recommended
 We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
- I want to create an IAM user
 We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

Autogenerated password
 You can view the password after you create the user.

Custom password
 Enter a custom password for the user.

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } | '

Show password

Users must create a new password at next sign-in - Recommended
 Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user.
[Learn more](#)

[Cancel](#) Next

Click on Attach Policies directly

Use AdministratorAccess Just for learning purpose

Permissions options

Add user to group
 Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
 Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly
 Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1168)
 Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS managed	0
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed	0
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	0

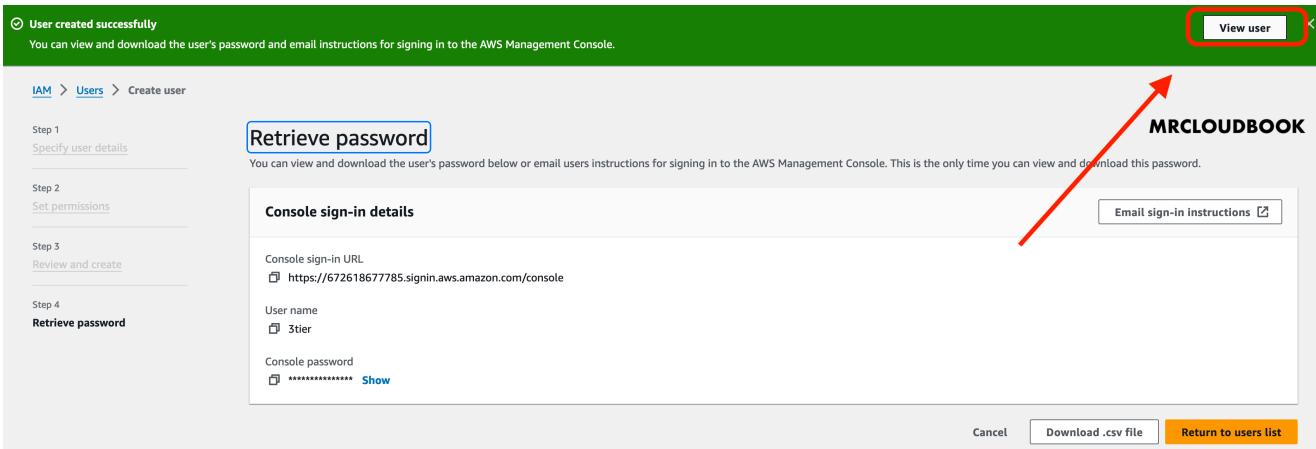
Click Next

[Cancel](#) Previous Next

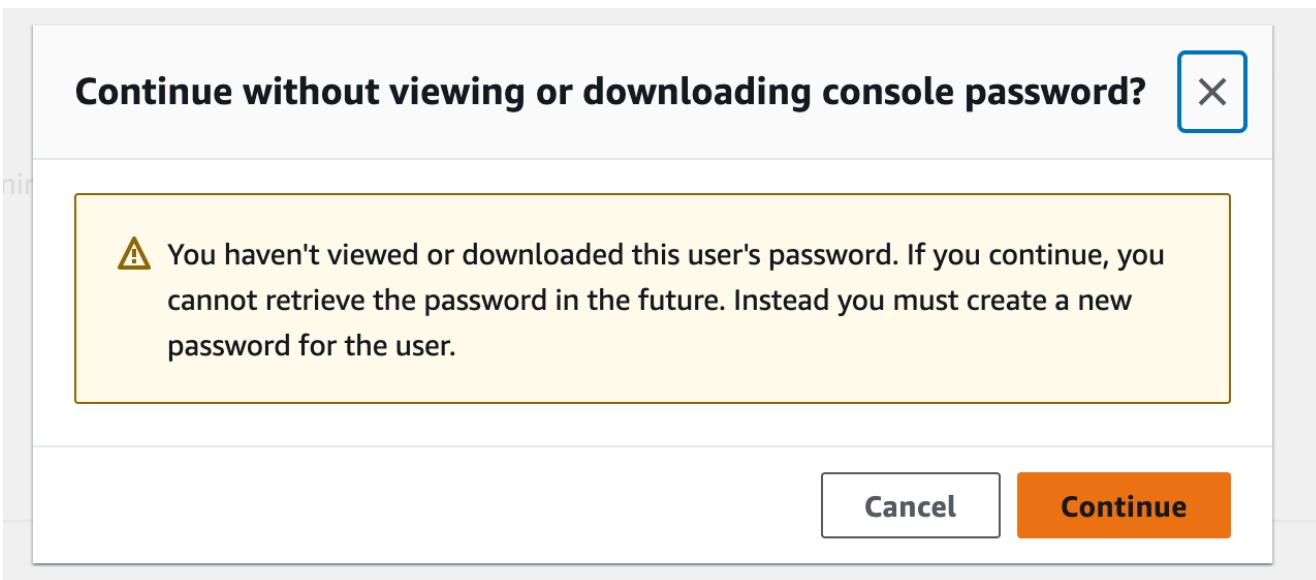
Click on Create user



Click on View user



Click on Continue



Click on Security credentials

IAM > Users > 3tier

3tier Info Delete

Summary		
ARN arn:aws:iam::672618677785:user/3tier	Console access ⚠ Enabled without MFA	Access key 1 Create access key
Created December 30, 2023, 16:02 (UTC+05:30)	Last console sign-in Never	
Permissions Groups Tags Security credentials Access Advisor		
MRCLLOUDBOOK		
Console sign-in Manage console access		
Console sign-in link https://672618677785.signin.aws.amazon.com/console	Console password Updated Now (2023-12-30 16:02 GMT+5:30)	Last console sign-in Never

Now under security credentials, Go to Access keys

Click on Create access key

Access keys (0) [Create access key](#)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

[Create access key](#)

Select CLI & Accept terms and click on Next

Use case

Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

Other
Your use case is not listed here.

Alternatives recommended

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

Confirmation

I understand the above recommendation and want to proceed to create an access key.

MR CLOUD BOOK

Cancel **Next**

Download .csv file and click on Done

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 AKIAZZGZSQYMZDKMUJUM	 ***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) **Done**

STEP2: Create EC2 Instance

- 1. Sign in to AWS Console:** Log in to your AWS Management Console.
- 2. Navigate to EC2 Dashboard:** Go to the EC2 Dashboard by selecting “Services” in the top menu and then choosing “EC2” under the Compute section.

- 3. Launch Instance:** Click on the “Launch Instance” button to start the instance creation process.
- 4. Choose an Amazon Machine Image (AMI):** Select an appropriate AMI for your instance. For example, you can choose Ubuntu image.
- 5. Choose an Instance Type:** In the “Choose Instance Type” step, select `t2.medium` as your instance type. Proceed by clicking “Next: Configure Instance Details.”
- 6. Configure Instance Details:**
 - For “Number of Instances,” set it to 1 (unless you need multiple instances).
 - Configure additional settings like network, subnets, IAM role, etc., if necessary.
 - For “Storage,” click “Add New Volume” and set the size to 8GB (or modify the existing storage to 16GB).
 - Click “Next: Add Tags” when you’re done.
- 7. Add Tags (Optional):** Add any desired tags to your instance. This step is optional, but it helps in organizing instances.
- 8. Configure Security Group:**
 - Choose an existing security group or create a new one.
 - Ensure the security group has the necessary inbound/outbound rules to allow access as required.
- 9. Review and Launch:** Review the configuration details. Ensure everything is set as desired.
- 10. Select Key Pair:**
 - Select “Choose an existing key pair” and choose the key pair from the dropdown.
 - Acknowledge that you have access to the selected private key file.
 - Click “Launch Instances” to create the instance.
- 11. Access the EC2 Instance:** Once the instance is launched, you can access it using the key pair and the instance’s public IP or DNS.

Ensure you have necessary permissions and follow best practices while configuring security groups and key pairs to maintain security for your EC2 instance.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	El...
3tier	i-02f6a895d63b86ca2	Running	t2.medium	Initializing	No alarms	ap-south-1a	ec2-13-200-255-119.ap...	13.200.255.119	-

Step3: Connect to Instance and Install Required Packages

```
Last login: Sat Dec 30 15:46:56 on ttys001
+ ~ cd Downloads/AJAY_1
+ AJAY_1 ssh -i "Mumbai-mac.pem" ubuntu@ec2-13-200-255-119.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-200-255-119.ap-south-1.compute.amazonaws.com (13.200.255.119)' can't be established.
ED25519 key fingerprint is SHA256::YXj0mcTc/6HMt9Pq/8U70FAdsFd+IpGqqSstqNAZAh0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-200-255-119.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Sat Dec 30 10:35:37 UTC 2023

 System load: 0.57373046875   Processes:           115
 Usage of /: 10.2% of 15.32GB  Users logged in:      0
 Memory usage: 5%            IPv4 address for eth0: 172.31.44.19
 Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-44-19:~$
```

Eksctl



```
sudo apt update
curl --silent --location "https://github.com/weaveworks/eksctl/releases"
sudo mv /tmp/eksctl /usr/local/bin
eksctl version
```



```
ubuntu@ip-172-31-44-19:~$ curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
p
MR CLOUD BOOK
ubuntu@ip-172-31-44-19:~$ sudo mv /tmp/eksctl /usr/local/bin
ubuntu@ip-172-31-44-19:~$ eksctl version
0.167.0
ubuntu@ip-172-31-44-19:~$
```

Kubectl



```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.3/2023-11-14,
chmod +x ./kubectl
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
kubectl version --client
```



```
ubuntu@ip-172-31-44-19:~$ curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.3/2023-11-14/bin/linux/amd64/kubectl
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total Spent  Left Speed
100 47.5M  100 47.5M    0     0  4294k      0  0:00:11  0:00:11  --:--:-- 5481k
ubuntu@ip-172-31-44-19:~$ █
```

```
ubuntu@ip-172-31-44-19:~$ sudo chmod +x ./kubectl
ubuntu@ip-172-31-44-19:~$ ls
kubectl
ubuntu@ip-172-31-44-19:~$ mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
ubuntu@ip-172-31-44-19:~$ MRCLOUDBOOK
ubuntu@ip-172-31-44-19:~$ kubectl version --client
Client Version: v1.28.3-eks-e71965b
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
ubuntu@ip-172-31-44-19:~$ █
```

Aws CLI



```
sudo apt install unzip -y
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```



```
ubuntu@ip-172-31-44-19:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total Spent  Left Speed
100 57.1M  100 57.1M    0     0  108M      0  --:--:--  --:--:--  --:--:-- 108M
```

```
ubuntu@ip-172-31-44-19:~$ aws --version
aws-cli/2.15.6 Python/3.11.6 Linux/6.2.0-1017-aws exe/x86_64/ubuntu.22 prompt/off
ubuntu@ip-172-31-44-19:~$ █
```

Helm

```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3  
chmod 700 get_helm.sh  
./get_helm.sh
```



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TFS curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TFS chmod 700 get_helm.sh  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TFS ./get_helm.sh  
Downloading https://get.helm.sh/helm-v3.13.1-linux-amd64.tar.gz  
Verifying checksum... Done.  
Preparing to install helm into /usr/local/bin  
helm installed into /usr/local/bin/helm  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TFS [ ]
```

STEP4: EKS Setup

Aws configure (Use us-east-1 region please)

```
aws configure
```



```
ubuntu@ip-172-31-44-19:~$ aws configure  
AWS Access Key ID [None]: AKIAZZGZ[REDACTED]JUM  
AWS Secret Access Key [None]: e87k9Axul+5XRdx[REDACTED]uDpnFnfQgyx9944y  
Default region name [None]: [REDACTED] us-east-1  
Default output format [None]: json  
ubuntu@ip-172-31-44-19:~$ [ ]
```

Lets clone GitHub repo

```
git clone https://github.com/Aj7Ay/3Tier-Robot-shop.git  
cd 3Tier-Robot-shop
```



```
ubuntu@ip-172-31-44-19:~$ git clone https://github.com/Aj7Ay/3Tier-Robot-shop.git
Cloning into '3Tier-Robot-shop'...
remote: Enumerating objects: 234, done.
remote: Counting objects: 100% (234/234), done.
remote: Compressing objects: 100% (186/186), done.
remote: Total 234 (delta 35), reused 233 (delta 35), pack-reused 0
Receiving objects: 100% (234/234), 16.70 MiB / 12.13 MiB/s, done.
Resolving deltas: 100% (35/35), done.
ubuntu@ip-172-31-44-19:~$ 
ubuntu@ip-172-31-44-19:~$ 
ubuntu@ip-172-31-44-19:~$ cd 3Tier-Robot-shop/
```

Create cluster



eksctl create cluster --name demo-cluster-three-tier-1 --region us-east-1



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ eksctl create cluster --name demo-cluster-three-tier-1 --region us-east-1
2023-12-30 11:37:43 [i] eksctl version 0.167.0
2023-12-30 11:37:43 [i] using region us-east-1      MRCLOUD BOOK
2023-12-30 11:37:44 [i] setting availability zones to [us-east-1b us-east-1d]
2023-12-30 11:37:44 [i] subnets for us-east-1b - public:192.168.0.0/19 private:192.168.64.0/19
2023-12-30 11:37:44 [i] subnets for us-east-1d - public:192.168.32.0/19 private:192.168.96.0/19
2023-12-30 11:37:44 [i] nodegroup "ng-8e645ae6" will use "" [AmazonLinux2/1.27]
2023-12-30 11:37:44 [i] using Kubernetes version 1.27
2023-12-30 11:37:44 [i] creating EKS cluster "demo-cluster-three-tier-1" in "us-east-1" region with managed nodes
2023-12-30 11:37:44 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-12-30 11:37:44 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=demo-cluster-three-tier-1'
2023-12-30 11:37:44 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "demo-cluster-three-tier-1" in "us-east-1"
2023-12-30 11:37:44 [i] CloudWatch logging will not be enabled for cluster "demo-cluster-three-tier-1" in "us-east-1"
2023-12-30 11:37:44 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=demo-cluster-three-tier-1'
2023-12-30 11:37:44 [i]
2 sequential tasks: { create cluster control plane "demo-cluster-three-tier-1",
  2 sequential sub-tasks: {
    wait for control plane to become ready,
    create managed nodegroup "ng-8e645ae6",
  }
}
```

Now Setup

Commands to configure IAM OIDC provider

USE CLUSTER NAME demo-cluster-three-tier-1



export cluster_name=<CLUSTER-NAME>



The command “export cluster_name=” is used in a computer’s command-line interface to create a named storage space (variable) that holds a specific value. It’s like giving a name to something so you can use it later. In this case, it’s creating a storage space called “cluster_name” and putting a value in it, which represents the name of a cluster. This helps remember and use the cluster’s name in other commands or programs without typing it repeatedly.



```
oidc_id=$(aws eks describe-cluster --name $cluster_name --query "cluster
```



This command uses the AWS CLI (Command Line Interface) to extract a specific piece of information about an Amazon EKS (Elastic Kubernetes Service) cluster

Check if there is an IAM OIDC provider configured already



```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```



This command utilizes the AWS CLI (Command Line Interface) to list OpenID Connect (OIDC) providers in your AWS Identity and Access Management (IAM) and extract specific information



```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name --approv
```



EKSCTL command used to associate the IAM OIDC provider with an Amazon EKS (Elastic Kubernetes Service) cluster.

```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ export cluster_name=demo-cluster-three-tier-1
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ aws eks describe-cluster --name $cluster_name --query "cluster.identity.oidc.issuer" --output text | cut -d '/' -f 5
An error occurred (ResourceNotFoundException) when calling the DescribeCluster operation: No cluster found for name: demo-cluster-three-tier-1.
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ aws configure
AWS Access Key ID [*****UJUM*]:
AWS Secret Access Key [*****944y*]:
Default region name [ap-south-1]: us-east-1
Default output format [json]:
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ oidc_id=$(aws eks describe-cluster --name $cluster_name --query "cluster.identity.oidc.issuer" --output text | cut -d '/' -f 5)
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ eksctl utils associate-iam-oidc-provider --cluster $cluster_name --approve
2023-12-30 11:54:47 [i] will create IAM Open ID Connect provider for cluster "demo-cluster-three-tier-1" in "us-east-1"
2023-12-30 11:54:48 [v] created IAM Open ID Connect provider for cluster "demo-cluster-three-tier-1" in "us-east-1"
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

MR CLOUD BOOK

Setup alb add on

Download IAM policy



```
curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/install/iam_policy.json
```



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ curl -O https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.5.4/docs/install/iam_policy.json
% Total    % Received % Xferd  Average Speed   Time      Time     Current
          Dload  Upload Total   Spent    Left Speed
100  8386  100  8386    0     0 23997      0 --:--:-- --:--:-- 24028
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

Create IAM Policy



```
aws iam create-policy \
--policy-name AWSLoadBalancerControllerIAMPolicy \
--policy-document file://iam_policy.json
```



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ aws iam create-policy \
--policy-name AWSLoadBalancerControllerIAMPolicy \
--policy-document file://iam_policy.json
{
  "Policy": {
    "PolicyName": "AWSLoadBalancerControllerIAMPolicy",
    "PolicyId": "ANPAZZGZSQYM2UNNR56KF",
    "Arn": "arn:aws:iam::672618677785:policy/AWSLoadBalancerControllerIAMPolicy",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2023-12-30T11:04:40+00:00",
    "UpdateDate": "2023-12-30T11:04:40+00:00"
  }
}
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

create IAM role

Please Add cluster name and Aws account ID

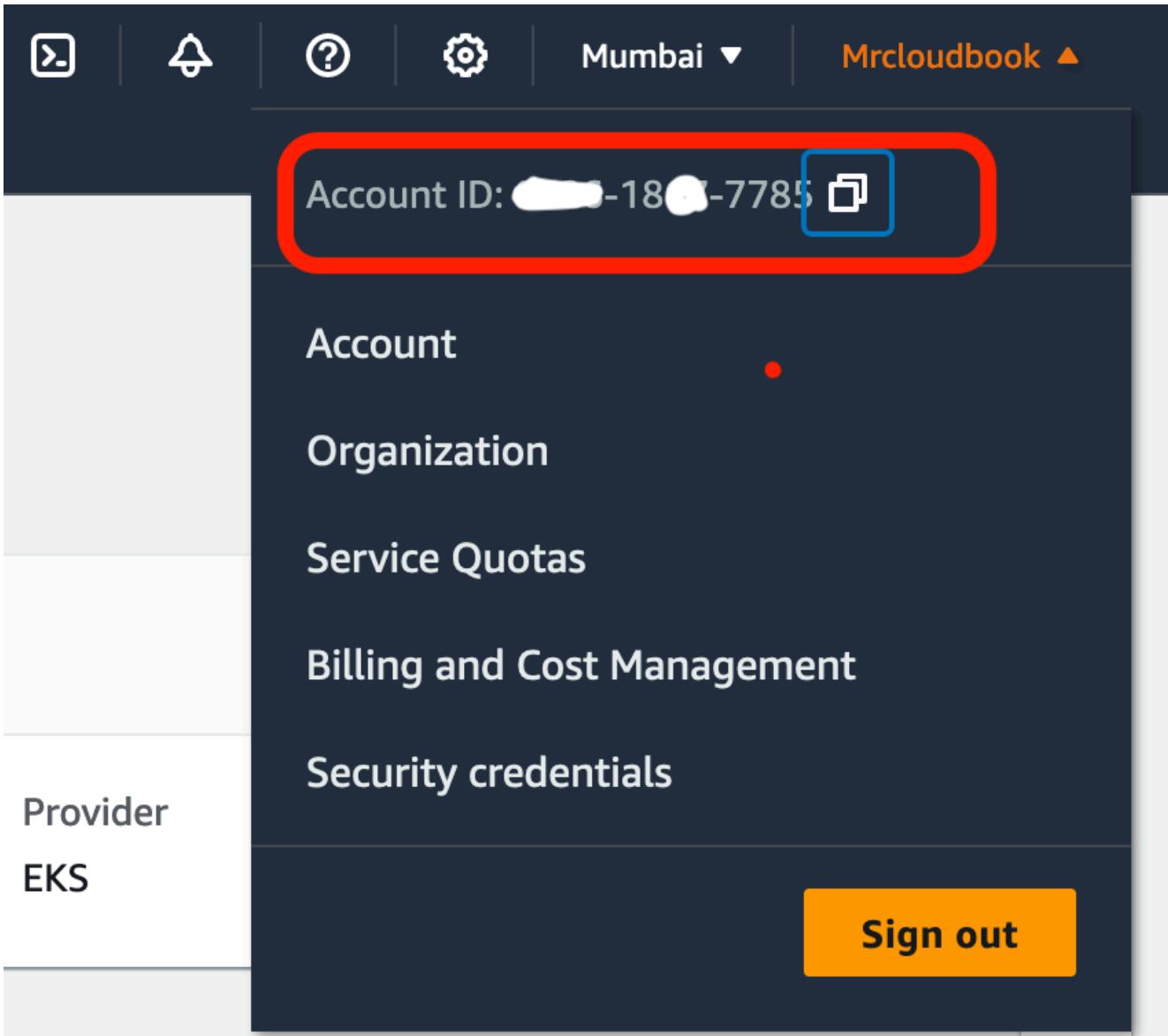


```
eksctl create iamserviceaccount \
--cluster=<your-cluster-name> \
--namespace=kube-system \
--name=aws-load-balancer-controller \
--role-name AmazonEKSLoadBalancerControllerRole \
--attach-policy-arn=arn:aws:iam::<your-aws-account-id>:policy/AWSLoadBalanc
--approve
```



To get aws account id

Go to aws console and click on Right side profile name and copy it



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ eksctl create iamserviceaccount \
--cluster=demo-cluster-three-tier-1 \
--namespace=kube-system \
--name=aws-load-balancer-controller \
--role-name AmazonEKSLoadBalancerControllerRole \
--attach-policy-arn=arn:aws:iam::[REDACTED]867[REDACTED]policy/AWSLoadBalancerControllerIAMPolicy \
--approve
2023-12-30 11:57:19 [i] 1 iamserviceaccount (kube-system/aws-load-balancer-controller) was included (based on the include/exclude rules)
2023-12-30 11:57:19 [!] serviceaccounts that exist in Kubernetes will be excluded, use --override-existing-serviceaccounts to override
2023-12-30 11:57:19 [i] 1 task: {
  2 sequential sub-tasks: {
    create IAM role for serviceaccount "kube-system/aws-load-balancer-controller",
    create serviceaccount "kube-system/aws-load-balancer-controller",
  } } 2023-12-30 11:57:19 [i] building iamserviceaccount stack "eksctl-demo-cluster-three-tier-1-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2023-12-30 11:57:19 [i] deploying stack "eksctl-demo-cluster-three-tier-1-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2023-12-30 11:57:19 [i] waiting for CloudFormation stack "eksctl-demo-cluster-three-tier-1-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2023-12-30 11:57:50 [i] waiting for CloudFormation stack "eksctl-demo-cluster-three-tier-1-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2023-12-30 11:57:51 [i] created serviceaccount "kube-system/aws-load-balancer-controller"
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

Deploy ALB controller

Add helm repo



```
helm repo add eks https://aws.github.io/eks-charts
```



Update the repo



```
helm repo update eks
```



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ helm repo add eks https://aws.github.io/eks-charts
"eks" has been added to your repositories
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ helm repo update eks
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "eks" chart repository
Update Complete. *Happy Helm-ing!
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

Install

please update VPC_ID in this command

go to eks and copy vpc id

EKS > Clusters > demo-cluster-three-tier-1

demo-cluster-three-tier-1

End of support for Kubernetes version 1.27 is July 2024. If you don't update your cluster to a later version before that date, it will automatically enter extended support. After the extended support preview ends, clusters on versions in extended support will be subject to additional fees. [Learn more](#) [Update now](#)

Cluster info [Info](#)

Status Active	Kubernetes version Info 1.27	Support type Standard support until July 2024	Provider EKS
-------------------------------	--	---	--------------

[Overview](#) [Resources](#) [Compute](#) **Networking** [Add-ons](#) [Access](#) [Observability](#) [Upgrade insights](#) [Update history](#) [Tags](#)

Networking

VPC Info vpc-05114ed1df2691116	Subnets subnet-0ddb7cff09d6868a1 (us-east-1b) subnet-045d45b53ec02aa42 (us-east-1d) subnet-0548b4368ba500381 (us-east-1b) subnet-0c5d5de9c08f29342 (us-east-1d)	Cluster security group Info sg-0468e23475df2319f	API server endpoint access Info Public
Cluster IP address family Info IPv4	Additional security groups sg-0b730d961518f3329	Public access source allowlist 0.0.0.0/0 (open to all traffic)	



```
helm install aws-load-balancer-controller eks/aws-load-balancer-control
```

```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ helm install aws-load-balancer-controller eks/aws-load-balancer-controller -n kube-system --set clusterName=demo-cluster-three-tier-1 --set serviceAccount.create=false --set serviceAccount.name=aws-load-balancer-controller --set region=us-east-1 --set vpcId=vpc-05114ed1df2691116
NAME: aws-load-balancer-controller
LAST DEPLOYED: Sat Dec 30 12:03:19 2023
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
AWS Load Balancer controller installed!
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

Verify that the deployments are running

```
kubectl get deployment -n kube-system aws-load-balancer-controller
```

```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ kubectl get deployment -n kube-system aws-load-balancer-controller
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
aws-load-balancer-controller   2/2      2          2           29s
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

EBS CSI Plugin configuration

The Amazon EBS CSI plugin requires IAM permissions to make calls to AWS APIs on your behalf.

Create an IAM role and attach a policy. AWS maintains an AWS managed policy or you can create your own custom policy. You can create an IAM role and attach the AWS managed policy with the following command. Replace my-cluster with the name of your cluster. The command deploys an AWS CloudFormation stack that creates an IAM role and attaches the IAM policy to it.

Please add Cluster name

```
eksctl create iamserviceaccount \
--name ebs-csi-controller-sa \
```

```
--namespace kube-system \
--cluster <YOUR-CLUSTER-NAME> \
--role-name AmazonEKS_EBS_CSI_DriverRole \
--role-only \
--attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCS \
--approve
```

MRCLOUDBOOK

```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ eksctl create iamserviceaccount \
--name ebs-csi-controller-sa \
--namespace kube-system \
--cluster demo-cluster-three-tier-1 \
--role-name AmazonEKS_EBS_CSI_DriverRole \
--role-only \
--attach-policy-arn arn:aws:iam::aws:policy/service-role/AmazonEBSCSIDriverPolicy \
--approve
2023-12-30 12:05:41 [i] 1 existing iamserviceaccount(s) (kube-system/aws-load-balancer-controller) will be excluded
2023-12-30 12:05:41 [i] 1 iamserviceaccount (kube-system/ebs-csi-controller-sa) was included (based on the include/exclude rules)
2023-12-30 12:05:41 [i] serviceaccounts in Kubernetes will not be created or modified, since the option --role-only is used
2023-12-30 12:05:41 [i] 1 task: { create IAM role for serviceaccount "kube-system/ebs-csi-controller-sa" }
2023-12-30 12:05:41 [i] building iamserviceaccount stack "eksctl-demo-cluster-three-tier-1-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2023-12-30 12:05:41 [i] deploying stack "eksctl-demo-cluster-three-tier-1-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2023-12-30 12:05:42 [i] waiting for CloudFormation stack "eksctl-demo-cluster-three-tier-1-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
2023-12-30 12:06:12 [i] waiting for CloudFormation stack "eksctl-demo-cluster-three-tier-1-addon-iamserviceaccount-kube-system-ebs-csi-controller-sa"
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

Run the following command. Replace with the name of your cluster, with your account ID.

```
eksctl create addon --name aws-ebs-csi-driver --cluster <YOUR-CLUSTER-NAME>
```

```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$ eksctl create addon --name aws-ebs-csi-driver --cluster demo-cluster-three-tier-1 --service-account-role-arn arn:aws:iam::672618677785:role/AmazonEKS_EBS_CSI_DriverRole --force
2023-12-30 12:07:40 [i] Kubernetes version "1.27" in use by cluster "demo-cluster-three-tier-1"
2023-12-30 12:07:41 [i] using provided ServiceAccountRoleARN "arn:aws:iam::672618677785:role/AmazonEKS_EBS_CSI_DriverRole"
2023-12-30 12:07:41 [i] creating addon
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS-TF$
```

Now Go inside the helm and create a namespace

```
cd helm
kubectl create ns robot-shop
```

```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS$  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS$ cd helm/  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ ls  
Chart.yaml README.md ingress.yaml templates values.yaml  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ kubectl create ns robot-shop  
namespace/robot-shop created  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ █
```

Now



```
helm install robot-shop --namespace robot-shop .
```



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ helm install robot-shop --namespace robot-shop .  
NAME: robot-shop  
LAST DEPLOYED: Sat Dec 30 12:10:08 2023  
NAMESPACE: robot-shop  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None  
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ █
```

Now check pods



```
kubectl get pods -n robot-shop
```



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ kubectl get pods -n robot-shop
NAME          READY   STATUS    RESTARTS   AGE
cart-dd947f945-qxb24   1/1     Running   0          5m3s
catalogue-7fd84d6c48-zs849  1/1     Running   0          5m3s
dispatch-66cb67494d-tcg2h   1/1     Running   0          5m3s
mongodb-db95c57c5-rjqmw   1/1     Running   0          5m3s
mysql-8cbc4749d-ptb99    1/1     Running   0          5m3s
payment-85f9dcf964-khgwm  1/1     Running   0          5m3s
rabbitmq-79488b58b5-qrhpns 1/1     Running   0          5m3s
ratings-5c79749c7b-snrrnh  1/1     Running   0          5m3s
redis-0                1/1     Running   0          5m3s
shipping-75f7cf6b6d-9jbcc  1/1     Running   0          5m3s
user-59745f7ccb-kd97c    1/1     Running   0          5m3s
web-656685795d-rcvdg    1/1     Running   0          5m3s
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$
```

Check service



kubectl get svc -n robot-shop



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ kubectl get svc -n robot-shop
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
cart      ClusterIP  10.100.30.133  <none>        8080/TCP    112s
catalogue ClusterIP  10.100.96.33   <none>        8080/TCP    112s
dispatch   ClusterIP  None           <none>        55555/TCP   112s
mongodb    ClusterIP  10.100.192.215 <none>        27017/TCP   112s
mysql      ClusterIP  10.100.212.97  <none>        3306/TCP    112s
payment    ClusterIP  10.100.69.251  <none>        8080/TCP    112s
rabbitmq   ClusterIP  10.100.64.122  <none>        5672/TCP,15672/TCP,4369/TCP 112s
ratings    ClusterIP  10.100.169.27  <none>        80/TCP      112s
redis      ClusterIP  10.100.109.206 <none>        6379/TCP   112s
shipping   ClusterIP  10.100.58.113  <none>        8080/TCP   112s
user       ClusterIP  10.100.235.240 <none>        8080/TCP   112s
web        LoadBalancer 10.100.236.39   k8s-robotshop-web-014176c61c-a1cc2a92b23b1625.elb.us-east-1.amazonaws.com  8080:30753/TCP 112s
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$
```

Now Apply ingress



kubectl apply -f ingress.yaml



```
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ ls
Chart.yaml README.md ingress.yaml templates values.yaml
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$ kubectl apply -f ingress.yaml
Warning: annotation "kubernetes.io/ingress.class" is deprecated, please use 'spec.ingressClassName' instead
ingress.networking.k8s.io/robot-shop created
ubuntu@ip-172-31-44-19:~/3Tier-Robot-shop/EKS/helm$
```

Now go to AWS CONSOLE
search for Ec2 and Go to load balancers

COPY DNS

EC2 > Load balancers > k8s-robotsho-robotsho-55094ff83e

Details

Load balancer type	Status	VPC	IP address type						
Application	Active	vpc-05114ed1df2691116	IPv4						
Scheme	Hosted zone	Availability Zones	Date created						
Internet-facing	Z35SXDOTRQ7X7K	subnet-0dbb7cf09d6868a1 us-east-1b (use1-az2) subnet-045d45b53ec02aa42 us-east-1d (use1-az6)	December 30, 2023, 17:45 (UTC+05:30)						
MRCLoudBook									
Load balancer ARN <code>arn:aws:elasticloadbalancing:us-east-1:672618677785:loadbalancer/app/k8s-robotsho-robotsho-55094ff83e/803e4a223fb0a0f9</code>									
DNS name info → COPY THIS <code>k8s-robotsho-robotsho-55094ff83e-302990556.us-east-1.elb.amazonaws.com (A Record)</code> <small>Copy DNS name of load balancer k8s-robotsho-robotsho-55094ff83e to clipboard</small>									
Listeners and rules Network mapping Security Monitoring Integrations Attributes Tags									
Tags (3) <input type="text" value="Filter tags"/> <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>elbv2.k8s.aws/cluster</td> <td>demo-cluster-three-tier-1</td> </tr> <tr> <td>ingress.k8s.aws/resource</td> <td>LoadBalancer</td> </tr> </tbody> </table>				Key	Value	elbv2.k8s.aws/cluster	demo-cluster-three-tier-1	ingress.k8s.aws/resource	LoadBalancer
Key	Value								
elbv2.k8s.aws/cluster	demo-cluster-three-tier-1								
ingress.k8s.aws/resource	LoadBalancer								

Open a new tab and paste

Stan's Robot Shop

Welcome to Stan's Robot Shop

Here you will find all of Stan's friends. Have a browse around and see who is here.

This is a simple example microservices ecommerce application. It has been built using various technologies:

- AngularJS (1.x)
- Nginx
- NodeJS
- Java
- Python
- Golang
- PHP (Apache)
- MongoDB
- Redis
- MySQL

When deployed into an environment monitored by Instana, these technology stacks will be automatically detected and monitored, all with minimum configuration. Every request will be traced end to end. Stan will keep an eye on all those metrics, events and traces and let you know what needs your attention.

To find out more visit the [Instana](#) site.

All the code is available on [Github](#).

MRCLoudBook

The screenshot shows the homepage of Stan's Robot Shop. On the left, there's a sidebar with links for Login/Register, Cart (Empty), and Categories (listing various robot models like Artificial Intelligence, Robot, Cybernetic Neutralization Android, etc.). The main area has two forms: one for logging in with fields for Name (Ajay) and Password (.....), and another for registering with fields for Name (Ajay), Email (mrcloudbook@gmail.com), Password (.....), Confirm Password (.....), and a Register button. A search bar at the top right contains the placeholder "Search". A small robot icon is in the top right corner.

STEP5: DELETE CLUSTER

JUST PROVIDE THIS COMMAND



```
eksctl delete cluster --name demo-cluster-three-tier-1 --region us-east-1
```



```
ubuntu@ip-172-31-44-19:~$ eksctl delete cluster --name demo-cluster-three-tier-1 --region us-east-1
2023-12-30 12:20:15 [i] deleting EKS cluster "demo-cluster-three-tier-1"
2023-12-30 12:20:17 [i] will drain 0 unmanaged nodegroup(s) in cluster "demo-cluster-three-tier-1"
2023-12-30 12:20:17 [i] starting parallel draining, max in-flight of 1
2023-12-30 12:20:18 [i] deleted 0 Fargate profile(s)
2023-12-30 12:20:21 [✓] kubeconfig has been updated
2023-12-30 12:20:21 [i] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
```

In conclusion, our journey through the deployment and configuration of Stan's Robot Shop, a versatile microservices application, has been an enlightening exploration into the world of containerized applications, orchestration, and monitoring.

Throughout this guide, we've covered a range of essential steps, from deploying the application using Docker Compose to associating IAM OIDC providers with Amazon EKS clusters, unlocking the potential for secure access to AWS resources through Kubernetes service accounts.

Stan's Robot Shop serves not only as a sandbox for experimenting with diverse technologies like NodeJS, Java, Python, and more but also as a practical learning ground for understanding orchestration tools like Kubernetes and monitoring solutions like Instana.

As you continue to delve into the intricacies of microservices architectures, container orchestration, and monitoring practices, remember that Stan's Robot Shop is an ideal starting point—a playground where you can further explore, test, and refine your skills in a safe and controlled environment.

We hope this guide has provided valuable insights and practical guidance, empowering you to take your knowledge and understanding of containerized applications and Kubernetes to the next level.



Ajay Kumar Yegireddi is a DevSecOps Engineer and System Administrator, with a passion for sharing real-world DevSecOps projects and tasks. **Mr. Cloud Book**, provides hands-on tutorials and practical insights to help others master DevSecOps tools and workflows. Content is designed to bridge the gap between development, security, and operations, making complex concepts easy to understand for both beginners and professionals.

Comments

9 responses to “Robot Shop: A Comprehensive Exploration of its 3-Tier Architecture, 8 Services, and 2 Databases”



soumya

1 January 2024

where is container & how it's achieving the CI process? if you can explain the flow it will be better. like, how to scale it with a container . inside it's repo there is one dockerfile for one microservice. how to use them etc etc. btw, thanks for your contribution.

[Reply](#)**mrcloudbook.com**

1 January 2024

Already i provided in link for the video in starting please watch if you have doubts

[Reply](#)**Mahmud Ghazni**

2 January 2024

Hi There,

I hope you're well! Your new website venture caught my eye, and I'm genuinely excited about its potential.

As a WordPress expert passionate about creating impactful digital experiences, I see great opportunities to elevate your project.

Could we schedule a quick chat to explore ideas? I'd love to share insights tailored to your goals. Let me know a time that suits you.

Looking forward to the possibility of collaborating!

Best,

Mahmud Ghazni

WordPress Expert Extraordinaire

Email: ghazni@itsyourdev.com

WhatsApp: <https://wa.me/8801322311024>

[Reply](#)**mrcloudbook.com**

2 January 2024

will think about it

[Reply](#)

**vishnu sai**

6 January 2024

can we make it to ci/cd setup and deploy it monitoring and checking alerts

[Reply](#)**mrcloobook.com**

6 January 2024

but it will be lengthy you can try bro if you want

[Reply](#)**Durga Bhavani**

23 May 2024

i followed the same process..why ingress file is not there in path ..is there any solution

[Reply](#)**mia**

17 June 2024

ingress file is in /EKS/helm/ingress.yaml

[Reply](#)**ajaychekurthy**

3 September 2024

why you change the dir while installing helm i didnt understand

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.



I'm not a robot

reCAPTCHA
Privacy - Terms

Post Comment

Uncategorized

How to Automate
Incident Response :
How Q Developer
Helped Me Automate a
Daily Pain Point

AI

How to Run Docker
Model Runner on
Ubuntu 24.04

11 July 2025

AI, DevOps

How to Install docker-
ai on Ubuntu 24.04

15 June 2025

22 July 2025

Upskill with Ajay: DevSecOps Mastery

Join Mr Cloud book to master DevSecOps through real-world projects. Learn CI/CD, security integration, automation, and more, gaining hands-on skills for industry-level challenges.



Important Links

[Privacy Policy](#)

[Terms & Conditions](#)

[Contact](#)

Resources

[Blog](#)

[YouTube Channel](#)

© 2024 · Powered by [Mr Cloud Book](#)

[Follow Us on YouTube](#)