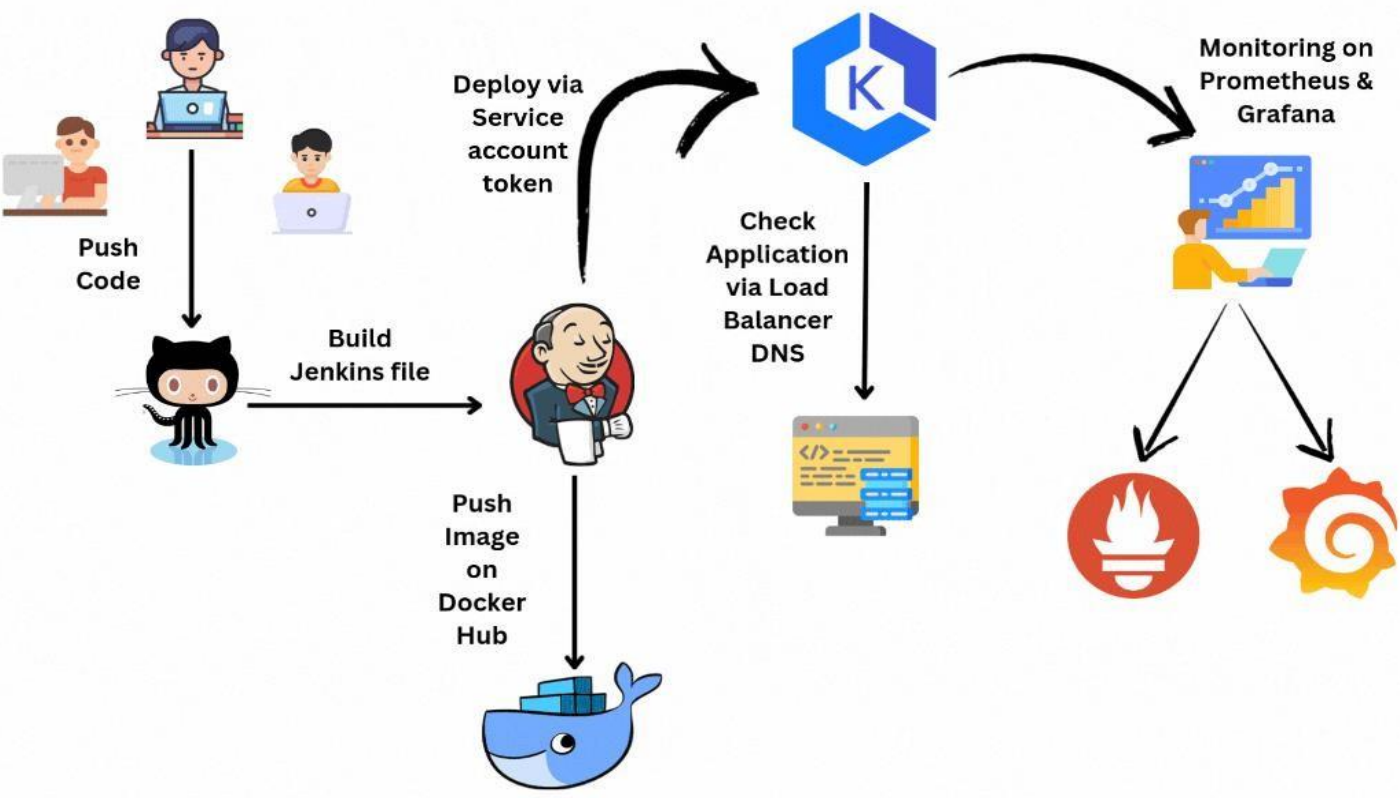


# PROJECT

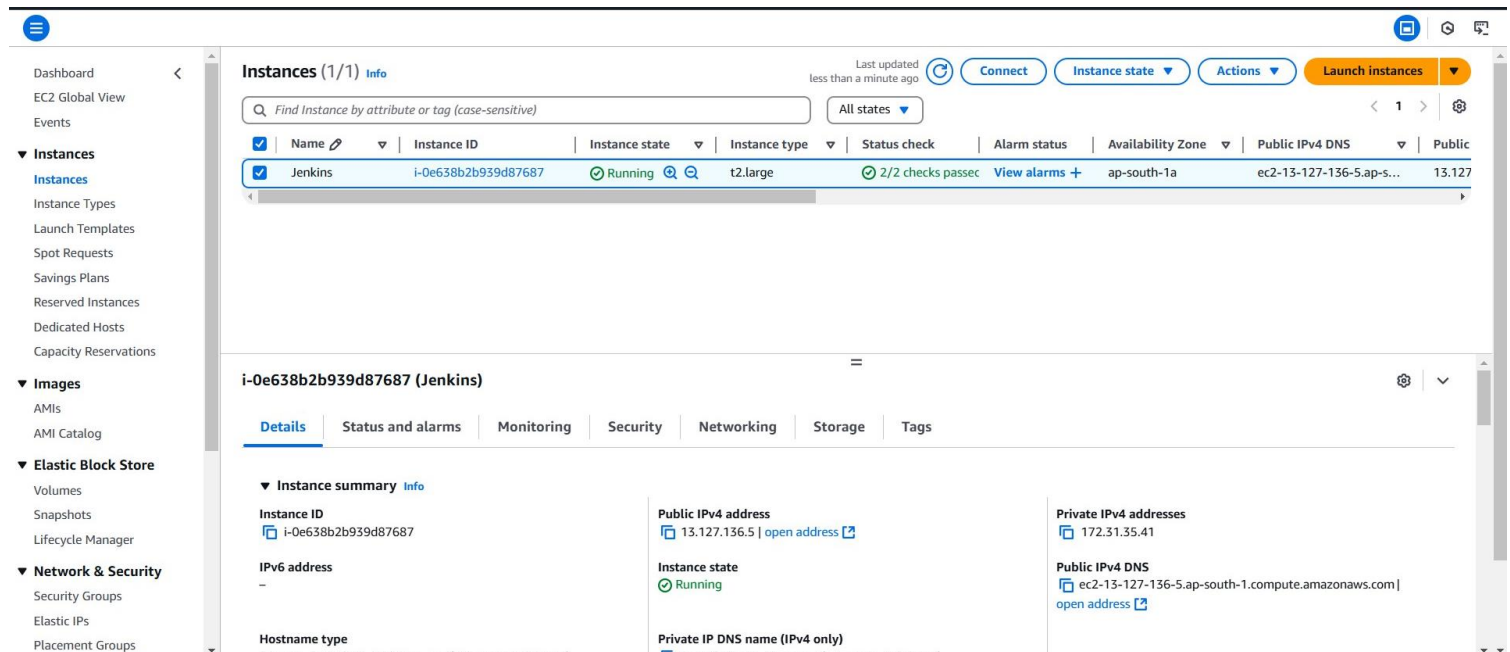
## End-to-End E-Commerce Microservices Built with AWS EKS, Jenkins, Docker, and Kubernetes

1. **AWS Console:** Used to manage servers and resources efficiently.
2. **AWS EKS (Elastic Kubernetes Service):** Facilitates the management of Kubernetes clusters on AWS.
3. **IAM (Identity and Access Management):** Manages user permissions and access to AWS resources.
4. **Jenkins:** Automates the CI/CD pipeline and application deployments.
5. **GitHub:** Hosts code repositories and provides version control.
6. **Docker:** Enables containerization of applications for simplified deployment and scaling.
7. **Docker Hub:** Serves as a platform to store and manage Docker images.
8. **kubectrl:** A command-line tool for interacting with Kubernetes clusters.
9. **eksctl:** Simplifies the creation and management of EKS clusters on AWS.
10. **Kubernetes:** An orchestration platform for automating the deployment, scaling, and management of containerized applications.
11. **Prometheus & Grafana:** Tools used for monitoring and visualizing system metrics.



## Step 1: Setting Up Your EC2 Instance

- **Storage:** Attach a 30 GB EBS volume.
- **Instance Type:** t2.large for better performance with 2 vCPUs and 8 GB RAM.
- **IAM Role:** Assign a role with full access to manage AWS resources.



## Step 2: Install AWS CLI, kubectl, and eksctl

### AWS CLI:

- **Installation:**

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

```
sudo apt install unzip
```

```
unzip awscliv2.zip
```

```
sudo ./aws/install
```

## **kubectrl:**

- **Installation:**

```
curl -o kubectrl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectrl
```

```
chmod +x ./kubectrl
```

```
sudo mv ./kubectrl /usr/local/bin
```

```
kubectrl version --short --client
```

## **eksctl:**

- **Installation:**

```
curl --silent --location
```

```
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

```
sudo mv /tmp/eksctl /usr/local/bin
```

```
eksctl version
```

## **STEP-3: CONFIGURE CREDENTIALS OF IAM USER**

➔ aws configure

```
root@ip-172-31-35-41:~# aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: ap-south-1
Default output format [None]: table
root@ip-172-31-35-41:~#
```

## STEP-4: CREATE CLUSTER

➤ **Cluster Creation:**

```
➔ eksctl create cluster --name=EKS-1 --region=ap-south-1 --zones=ap-south-1a,ap-south-1b --without-nodegroup
```

```

root@ip-172-31-35-41:~# eksctl create cluster --name=EKS-1 --region=ap-south-1 --zones=ap-south-1a,ap-south-1b --without-nodegroup
2025-01-01 08:24:50 [i] eksctl version 0.199.0
2025-01-01 08:24:50 [i] using region ap-south-1
2025-01-01 08:24:50 [i] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2025-01-01 08:24:50 [i] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2025-01-01 08:24:50 [i] using Kubernetes version 1.30
2025-01-01 08:24:50 [i] creating EKS cluster "EKS-1" in "ap-south-1" region with
2025-01-01 08:24:50 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=EKS-1'
2025-01-01 08:24:50 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "EKS-1" in "ap-south-1"
2025-01-01 08:24:50 [i] CloudWatch logging will not be enabled for cluster "EKS-1" in "ap-south-1"
2025-01-01 08:24:50 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=EKS-1'
2025-01-01 08:24:50 [i] default addons vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2025-01-01 08:24:50 [i]
2 sequential tasks: { create cluster control plane "EKS-1",
  2 sequential sub-tasks: {
    1 task: { create addons },
    wait for control plane to become ready,
  }
}
2025-01-01 08:24:50 [i] building cluster stack "eksctl-EKS-1-cluster"
2025-01-01 08:24:51 [i] deploying stack "eksctl-EKS-1-cluster"
2025-01-01 08:25:21 [i] waiting for CloudFormation stack "eksctl-EKS-1-cluster"
2025-01-01 08:25:51 [i] waiting for CloudFormation stack "eksctl-EKS-1-cluster"
2025-01-01 08:26:51 [i] waiting for CloudFormation stack "eksctl-EKS-1-cluster"
2025-01-01 08:27:51 [i] waiting for CloudFormation stack "eksctl-EKS-1-cluster"
2025-01-01 08:28:51 [i] waiting for CloudFormation stack "eksctl-EKS-1-cluster"
2025-01-01 08:29:51 [i] waiting for CloudFormation stack "eksctl-EKS-1-cluster"
2025-01-01 08:30:51 [i] waiting for CloudFormation stack "eksctl-EKS-1-cluster"
2025-01-01 08:31:51 [i] waiting for CloudFormation stack "eksctl-EKS-1-cluster"
2025-01-01 08:31:52 [!] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2025-01-01 08:31:52 [i] creating addon
2025-01-01 08:31:52 [i] successfully created addon
2025-01-01 08:31:52 [i] creating addon
2025-01-01 08:31:52 [i] successfully created addon
2025-01-01 08:31:52 [i] creating addon
2025-01-01 08:31:53 [i] successfully created addon
2025-01-01 08:33:53 [i] waiting for the control plane to become ready
2025-01-01 08:33:54 [✓] saved kubeconfig as "/root/.kube/config"
2025-01-01 08:33:54 [i] no tasks
2025-01-01 08:33:54 [✓] all EKS cluster resources for "EKS-1" have been created
2025-01-01 08:33:55 [i] kubectrl command should work with "/root/.kube/config", try 'kubectrl get nodes'
2025-01-01 08:33:55 [✓] EKS cluster "EKS-1" in "ap-south-1" region is ready
root@ip-172-31-35-41:~#

```

☰

EKS > Clusters > EKS-1

Amazon Elastic Kubernetes Service

Clusters

▼ Amazon EKS Anywhere

Enterprise Subscriptions New

▼ Related services

Amazon ECR

AWS Batch

Console settings

Documentation

Submit feedback

EKS-1

🔄

Delete cluster

Upgrade version

View dashboard

🔔

Your current IAM principal doesn't have access to Kubernetes objects on this cluster.  
This may be due to the current user or role not having Kubernetes RBAC permissions to describe cluster resources or not having an entry in the cluster's auth config map. [Learn more](#)

🔔

End of standard support for Kubernetes version 1.30 is July 28, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the [pricing page](#).

Upgrade now

▼ Cluster info Info

Status  
🟢 Active

Cluster health issues  
🟢 0

Kubernetes version Info  
1.30

Upgrade insights  
🟢 4 🟡 1

Support period  
🔔 Standard support until July 28, 2025

Node health issues  
🟢 0

Provider  
EKS

Overview

Resources

Compute

Networking

Add-ons 1

Access

Observability

Update history

Tags

Details

API server endpoint  
🔗 <https://7731EE3764A191763BFB342FB9845451.yl4.ap-sou-th-1.eks.amazonaws.com>

Certificate authority  
🔗 [LSOHL51CRUDJTIBDRVJUSUZJQOFURSOtLSOck1JSU](#)

OpenID Connect provider URL  
🔗 <https://oidc.eks.ap-south-1.amazonaws.com/id/7731EE3764A191763BFB342FB9845451>

Cluster IAM role ARN  
🔗 [arn:aws:iam::339713120000:role/eksctl-EKS-1-cluster-Servi ceRole-FMSJgv8Rdh8T](#)

Created  
🔗 42 minutes ago

Cluster ARN  
🔗 [arn:aws:eks:ap-south-1:339713120000:cluster/EKS-1](#)

Platform version Info

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)



➤ **OIDC Provider:**

➔ eksctl utils associate-iam-oidc-provider --region ap-southeast-1 --cluster EKS-1 --approve

```
root@ip-172-31-35-41:~# eksctl utils associate-iam-oidc-provider --region ap-south-1 --cluster EKS-1 --approve
2025-01-01 08:34:54 [i] will create IAM Open ID Connect provider for cluster "EKS-1" in "ap-south-1"
2025-01-01 08:34:54 [v] created IAM Open ID Connect provider for cluster "EKS-1" in "ap-south-1"
root@ip-172-31-35-41:~#
```

➤ **Node Group Creation:**

➔ eksctl create nodegroup --cluster=EKS-1 --region=ap-southeast-1 --name=node2 --node-type=t3.medium --nodes=3 --nodes-min=2 --nodes-max=4 --node-volume-size=20 --ssh-access --ssh-public-key=DevOps --managed --asg-access --external-dns-access --full-ecr-access --appmesh-access --alb-ingress-access

```
root@ip-172-31-35-41:~# eksctl create nodegroup --cluster=EKS-1 --region=ap-south-1 --name=node2 --node-type=t3.medium --nodes=3 --nodes-min=2 --nodes-max=4 --node-volume-size=20 --ssh-access --ssh-public-key=rds --managed --asg-access --external-dns-access --full-ecr-access --appmesh-access --alb-ingress-access
2025-01-01 08:36:28 [i] will use version 1.30 for new nodegroup(s) based on control plane version
2025-01-01 08:36:28 [i] nodegroup "node2" will use "" [AmazonLinux2/1.30]
2025-01-01 08:36:28 [i] using EC2 key pair "rds"
2025-01-01 08:36:29 [i] 1 nodegroup (node2) was included (based on the include/exclude rules)
2025-01-01 08:36:29 [i] will create a CloudFormation stack for each of 1 managed nodegroups in cluster "EKS-1"
2025-01-01 08:36:29 [i]
2 sequential tasks: { fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "node2" } } }
2025-01-01 08:36:29 [i]
2025-01-01 08:36:29 [i] checking cluster stack for missing resources
2025-01-01 08:36:29 [i] cluster stack has all required resources
2025-01-01 08:36:29 [i] building managed nodegroup stack "eksctl-EKS-1-nodegroup-node2"
2025-01-01 08:36:29 [i] deploying stack "eksctl-EKS-1-nodegroup-node2"
2025-01-01 08:36:29 [i] waiting for CloudFormation stack "eksctl-EKS-1-nodegroup-node2"
2025-01-01 08:36:29 [i] waiting for CloudFormation stack "eksctl-EKS-1-nodegroup-node2"
2025-01-01 08:36:29 [i] waiting for CloudFormation stack "eksctl-EKS-1-nodegroup-node2"
2025-01-01 08:37:54 [i] waiting for CloudFormation stack "eksctl-EKS-1-nodegroup-node2"
2025-01-01 08:39:39 [i] waiting for CloudFormation stack "eksctl-EKS-1-nodegroup-node2"
2025-01-01 08:39:39 [i] no tasks
2025-01-01 08:39:39 [v] created 0 nodegroup(s) in cluster "EKS-1"
2025-01-01 08:39:39 [i] nodegroup "node2" has 3 node(s)
2025-01-01 08:39:39 [i] node "ip-192-168-26-74.ap-south-1.compute.internal" is ready
2025-01-01 08:39:39 [i] node "ip-192-168-56-251.ap-south-1.compute.internal" is ready
2025-01-01 08:39:39 [i] node "ip-192-168-56-39.ap-south-1.compute.internal" is ready
2025-01-01 08:39:39 [i] waiting for at least 2 node(s) to become ready in "node2"
2025-01-01 08:39:39 [i] nodegroup "node2" has 3 node(s)
2025-01-01 08:39:39 [i] node "ip-192-168-26-74.ap-south-1.compute.internal" is ready
2025-01-01 08:39:39 [i] node "ip-192-168-56-251.ap-south-1.compute.internal" is ready
2025-01-01 08:39:39 [i] node "ip-192-168-56-39.ap-south-1.compute.internal" is ready
2025-01-01 08:39:39 [v] created 1 managed nodegroup(s) in cluster "EKS-1"
2025-01-01 08:39:39 [i] checking security group configuration for all nodegroups
2025-01-01 08:39:39 [i] all nodegroups have up-to-date cloudformation templates
root@ip-172-31-35-41:~#
```

The screenshot shows the AWS Management Console interface for the 'Instances' page. The left sidebar contains navigation links for Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, and Placement Groups. The main content area shows a list of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IP v4 DNS, and Public. The 'Monitoring' tab is selected, displaying four graphs: CPU utilization (%), Network in (bytes), Network out (bytes), and Network packets in (count). The 'Launch wizard' link is visible in the top right corner.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP v4 DNS	Public
EKS-1-node2-...	i-0accda8f7ddae530b	Running	t3.medium	3/3 checks passed	View alarms +	ap-south-1a	ec2-3-109-216-195.ap-...	3.109.127
Jenkins	i-0e638b2b939d87687	Running	t2.large	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-127-136-5.ap-s...	13.127
EKS-1-node2-...	i-0e06fbb25dd0c7261	Running	t3.medium	3/3 checks passed	View alarms +	ap-south-1b	ec2-13-201-188-147.ap...	13.201
EKS-1-node2-...	i-010b2aeacd0880440	Running	t3.medium	3/3 checks passed	View alarms +	ap-south-1b	ec2-3-110-37-238.ap-s...	3.110.127

## STELP-5: INSTALL JENKINS & DOCKER

### ➤ Jenkins:

- ➔ sudo apt install openjdk-17-jre-headless -y
- sudo wget -O /usr/share/keyrings/jenkins-keyring.asc  
<https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key>  
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]  
<https://pkg.jenkins.io/debian-stable> binary/" | sudo tee  
/etc/apt/sources.list.d/jenkins.list > /dev/null
- sudo apt-get update
- sudo apt-get install jenkins -y

```
root@ip-172-31-35-41:~# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-01-01 08:58:14 UTC; 1min 22s ago
     Main PID: 4629 (java)
        Tasks: 45 (limit: 9507)
      Memory: 1000.1M (peak: 1001.1M)
         CPU: 17.728s
        CGroup: /system.slice/jenkins.service
               └─4629 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Jan 01 08:58:09 ip-172-31-35-41 jenkins[4629]: 56b43e9d5ab744e79a1016dff7eaffd2
Jan 01 08:58:09 ip-172-31-35-41 jenkins[4629]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Jan 01 08:58:09 ip-172-31-35-41 jenkins[4629]: *****
Jan 01 08:58:09 ip-172-31-35-41 jenkins[4629]: *****
Jan 01 08:58:09 ip-172-31-35-41 jenkins[4629]: *****
Jan 01 08:58:14 ip-172-31-35-41 jenkins[4629]: 2025-01-01 08:58:14.075+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: C
Jan 01 08:58:14 ip-172-31-35-41 jenkins[4629]: 2025-01-01 08:58:14.091+0000 [id=23] INFO hudson.lifecycle.Lifecycle#onReady: Jenki
Jan 01 08:58:14 ip-172-31-35-41 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Jan 01 08:58:15 ip-172-31-35-41 jenkins[4629]: 2025-01-01 08:58:15.575+0000 [id=48] INFO h.m.DownloadService$Downloadable#load: Ob
Jan 01 08:58:15 ip-172-31-35-41 jenkins[4629]: 2025-01-01 08:58:15.575+0000 [id=48] INFO hudson.util.Retrier#start: Performed the
lines 1-20/20 (END)

root@ip-172-31-35-41:~# cat /var/lib/jenkins/secrets/initialAdminPassword
56b43e9d5ab744e79a1016dff7eaffd2
root@ip-172-31-35-41:~#
```

Dashboard >

+ New Item

📁 Build History

⚙️ Manage Jenkins

📌 My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

✎ Add description

### Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

+

Set up a distributed build

Set up an agent

📄

Configure a cloud

☁

Learn more about distributed builds

?

REST API

Jenkins 2.479.2

## ➤ Docker:

➔ sudo apt install docker.io -y

➔ sudo chmod 777 /var/run/docker.sock

```
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-35-41:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-01-01 09:04:28 UTC; 38s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 5348 (dockerd)
      Tasks: 9
     Memory: 28.8M (peak: 29.1M)
        CPU: 263ms
    CGroup: /system.slice/docker.service
            └─5348 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 01 09:04:28 ip-172-31-35-41 systemd[1]: Starting docker.service - Docker Application Container Engine...
Jan 01 09:04:28 ip-172-31-35-41 dockerd[5348]: time="2025-01-01T09:04:28.601079004Z" level=info msg="Starting up"
Jan 01 09:04:28 ip-172-31-35-41 dockerd[5348]: time="2025-01-01T09:04:28.602470409Z" level=info msg="detected 127.0.0.53 nameserver, assuming sy
Jan 01 09:04:28 ip-172-31-35-41 dockerd[5348]: time="2025-01-01T09:04:28.701271232Z" level=info msg="Loading containers: start."
Jan 01 09:04:28 ip-172-31-35-41 dockerd[5348]: time="2025-01-01T09:04:28.926393013Z" level=info msg="Loading containers: done."
Jan 01 09:04:28 ip-172-31-35-41 dockerd[5348]: time="2025-01-01T09:04:28.943002350Z" level=info msg="Docker daemon" commit="26.1.3-0ubuntu1~24.0
Jan 01 09:04:28 ip-172-31-35-41 dockerd[5348]: time="2025-01-01T09:04:28.943140297Z" level=info msg="Daemon has completed initialization"
Jan 01 09:04:28 ip-172-31-35-41 dockerd[5348]: time="2025-01-01T09:04:28.995757768Z" level=info msg="API listen on /run/docker.sock"
Jan 01 09:04:28 ip-172-31-35-41 systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

## STEP-6: DOWNLOAD PLUGINS

### ➤ Required Plugins:

➔ Docker

➔ Docker Pipeline

➔ Kubernetes

➔ Kubernetes CLI

Dashboard > Manage Jenkins > Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Search available plugins

Install

Install	Name	Released
<input checked="" type="checkbox"/>	<div>Docker 1.7.0</div> <div>Cloud Providers Cluster Management docker</div> <div>This plugin integrates Jenkins with Docker</div>	2 mo 18 days ago
<input checked="" type="checkbox"/>	<div>Docker Pipeline 580.vc0c340686b_54</div> <div>pipeline DevOps Deployment docker</div> <div>Build and use Docker containers from pipelines.</div>	7 mo 14 days ago
<input checked="" type="checkbox"/>	<div>Kubernetes 4306.vc91e951ea_eb_d</div> <div>Cloud Providers Cluster Management kubernetes Agent Management</div> <div>This plugin integrates Jenkins with Kubernetes</div>	12 days ago
<input checked="" type="checkbox"/>	<div>Kubernetes CLI 1.12.1</div> <div>kubernetes</div> <div>Configure kubectl for Kubernetes</div>	1 yr 4 mo ago



## Configure Jenkins for Docker:

### ➤ In Jenkins Dashboard:

➔ Manage Jenkins ➔ Tools ➔ Docker installations ➔ Name: docker ➔ Install automatically ➔ Docker version: latest.

Dashboard > Manage Jenkins > Tools

Docker installations

Add Docker

⌵ Docker

Name

docker

☒ Install automatically ?

⌵ Download from docker.com

Docker version ?

latest

Add Installer ▾

Add Docker

Save Apply

## STEP-7: ADD CREDentials FOR DOCKER & GITHUB

### Add DockerHub Credentials:

- Dashboard ➔ Manage Jenkins ➔ Credentials ➔ System ➔ Global credentials (unrestricted).
- Add: Username: lalitmahajan, Password: [Docker password], ID: docker.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind

Username with password ▾

Scope ?

Global (Jenkins, nodes, items, all child items, etc) ▾

Username ?

lalitmahajan

☐ Treat username as secret ?

Password ?

.....

ID ?

docker-cred

Description ?

Create

## STEP-8: CREATE SERVICE ACCOUNT, ROLE, ROLE BIND FOR webapps Namespace

→ NameSpace

```
apiVersion: v1
kind: Namespace
metadata:
  name: webapps
```

```
root@ip-172-31-35-41:~# mkdir manifest
root@ip-172-31-35-41:~# cd manifest
root@ip-172-31-35-41:~/manifest# vim namespace.yml
root@ip-172-31-35-41:~/manifest# kubectl create -f namespace.yml
namespace/webapps created
root@ip-172-31-35-41:~/manifest# kubectl get ns
```

NAME	STATUS	AGE
default	Active	54m
kube-node-lease	Active	54m
kube-public	Active	54m
kube-system	Active	54m
webapps	Active	19s

```
root@ip-172-31-35-41:~/manifest# |
```

→ Service Account

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins
  namespace: webapps
```

```
root@ip-172-31-35-41:~/manifest# vim svc-acc.yml
root@ip-172-31-35-41:~/manifest# kubectl create -f svc-acc.yml
serviceaccount/jenkins created
root@ip-172-31-35-41:~/manifest# kubectl get sa
```

NAME	SECRETS	AGE
default	0	56m

```
root@ip-172-31-35-41:~/manifest# kubectl get sa -n webapps
```

NAME	SECRETS	AGE
default	0	2m21s
jenkins	0	34s

```
root@ip-172-31-35-41:~/manifest# |
```

## → Role

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: app-role
  namespace: webapps
rules:
  - apiGroups:
      - ""
      - apps
      - autoscaling
      - batch
      - extensions
      - policy
      - rbac.authorization.k8s.io
    resources:
      - pods
      - componentstatuses
      - configmaps
      - daemonsets
      - deployments
      - events
      - endpoints
      - horizontalpodautoscalers
      - ingress
      - jobs
      - limitranges
      - namespaces
      - nodes
      - pods
      - persistentvolumes
      - persistentvolumeclaims
      - resourcequotas
      - replicaset
      - replicationcontrollers
      - serviceaccounts
      - services
    verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: app-rolebinding
  namespace: webapps
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: app-role
subjects:
  - namespace: webapps
    kind: ServiceAccount
    name: jenkins |
-- INSERT --
```

```
root@ip-172-31-35-41:~/manifest# vim role.yml
root@ip-172-31-35-41:~/manifest# kubectl create -f role.yml
role.rbac.authorization.k8s.io/app-role created
rolebinding.rbac.authorization.k8s.io/app-rolebinding created
root@ip-172-31-35-41:~/manifest#
```

## → Token

```
apiVersion: v1
kind: Secret
type: kubernetes.io/service-account-token
metadata:
  name: mysecretname
  namespace: webapps
  annotations:
    kubernetes.io/service-account.name: jenkins
```

```
root@ip-172-31-35-41:~/manifest#  
root@ip-172-31-35-41:~/manifest# vim token.yml  
root@ip-172-31-35-41:~/manifest# kubectl create -f token.yml  
secret/mysecretname created  
root@ip-172-31-35-41:~/manifest#
```

## Get Token:

- **Command:**

➔ `kubectl describe secret mysecretname -n webapps`

```
root@ip-172-31-35-41:~/manifest# kubectl describe secret mysecretname -n webapps
Name:          mysecretname
Namespace:     webapps
Labels:        <none>
Annotations:   kubernetes.io/service-account.name: jenkins
                kubernetes.io/service-account.uid: b7b54b94-2a49-4d7c-94dd-dde452674dc8

Type: kubernetes.io/service-account-token

Data
====
ca.crt:      1107 bytes
namespace:   7 bytes
token:       eyJhbGciOiJSUzI1NiIsImtpZCI6IlNvcmRlRTVkc0StLWcEseUpmQUZIeHvyelNvLXJfcmRjHX0ZXaffKdloySm8ifQ.eyJpc3MiOiJrdwJlcnM5ldGvZlNlcnZpY2VhY2NdvdW50Iiwia3ViZXJuzXRlcyspbpy9zzXJ2awNlYWwjb3VudC9yUWY1lzc3BhY2UiOiJ3ZWJhcHBzIiwia3ViZXJuzXRlcyspbpy9zzXJ2awNlYWwjb3VudC9zzWNyZXQubmFtZSI6Im15c2VjcmV0bmFtZSIsImt1YmVybmV0ZXMuaW8vc2VydmIdjWZWFjY291bnQvc2VydmIdjZSlhy2NdvdW50Lm5hbWUioiJqZw5raw5ziiwia3ViZXJuzXRlcyspbpy9zzXJ2awNlYWwjb3VudC9zzXJ2awNlWFJy291bnQuZDdlk1IjoieYjdINTRiotQtMmEOOS00ZbdjLTk0ZGQtZGRlNDUyNjc0ZGM4Iiwic3ViIjoieGlzdGwtbnNlcnZpY2VhY2NdvdW50ondlYWwMFWcHM6Mvua2lucyJ9.YNVc_cu0VEstEkCae7vHmL9XQGqIfZ6qxUjoeEVIVU9MgiLHuyHBkt2r8xcMLZAqTE0WvgWVCbfCrTcf-qIYu1toBuBMtgbvxxKGJwZ6gMpBWwzG8ZwtmdmPTnlgsbaVEE2tD0dttnQyDnqt6e-rIWhuE5gcEG0FXH1ZZqZD4XX0Kcuovdpvuq0vkLVgcrcH-bstfsgrMB2xSY1tts_4nfsIYBRUKWDtdgmUnc-sFEI9M7PCsx1uh_sur-hH8RoRNTGGiqll_4hNdHva-8oOh_D07qcQ02BAzzqOWniPq2qDow3CHPt37EOur5TAox3scK7nh61IoGoKePhwjzy-g
```

## STEP-9: ADD TOKEN TO CRED

### Add Token:

- Dashboard → Manage Jenkins → Credentials → System → Global credentials (unrestricted).
- Add: Secret: [Generated token], ID: k8s-token.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

### New credentials

Kind

Secret text



Scope ?

Global (Jenkins, nodes, items, all child items, etc)



Secret

.....

ID ?

k8s-token

Description ?

Create



## Configuration

## General

## Branch Sources

## Build Configuration

## Scan Multibranch Pipeline Triggers


## Orphaned Item Strategy

## Appearance

## Health metrics

## Properties

## General


Enabled Display Name 

Description

Plain text [Preview](#)

## Branch Sources

## Git

Project Repository Credentials 

Save

Apply

## Build Configuration

Mode

Script Path 

## Scan Multibranch Pipeline Triggers

☐ Periodically if not otherwise run 

## Orphaned Item Strategy

Jobs for removed SCM heads (i.e. deleted branches) can be removed immediately or kept based on a desired retention strategy. By default, jobs will be removed as soon as Jenkins determines their associated SCM head no longer exists. As an example, it may be useful to configure a different retention strategy to be able to examine build results of a branch after it has been removed.

☐ Abort builds ☒ Discard old items

Days to keep old items

if not empty, old items are only kept up to this number of days

Save

Apply

Status

Configure

Scan Multibranch Pipeline Now

Scan Multibranch Pipeline Log

Multibranch Pipeline Events

Delete Multibranch Pipeline

Build History

Project Relationship

Check File Fingerprint

Rename

Pipeline Syntax

Credentials

Build Queue

Build Executor Status0/4

Micro-Service-Project

Branches (12)

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	adservice	15 min #1	N/A	3 min 22 sec
✓	☀	cartservice	15 min #1	N/A	3 min 16 sec
✓	☀	checkoutservice	15 min #1	N/A	4 min 18 sec
✓	☀	currencyservice	15 min #1	N/A	7 min 12 sec
✓	☀	emailservice	15 min #1	N/A	7 min 51 sec
✓	☀	frontend	15 min #1	N/A	6 min 1 sec
✓	☀	loadgenerator	15 min #1	N/A	5 min 30 sec
✓	☁	main	10 sec #7	4 min 3 sec #6	4.3 sec
✓	☀	paymentservice	15 min #1	N/A	7 min 53 sec
✓	☀	productcatalogservice	15 min #1	N/A	5 min 55 sec
✓	☀	recommendationservice	15 min #1	N/A	7 min 59 sec
✓	☀	shippingservice	15 min #1	N/A	4 min 21 sec

# Load Balancer

EC2 > Load balancers > a2caf46d4d2e34512b60a74b4968f9de

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

Load Balancers

Target Groups

Trust StoresNew

Auto Scaling

Auto Scaling Groups

a2caf46d4d2e34512b60a74b4968f9de

Actions

▼ Details

Load balancer type

Classic

Status

3 of 3 instances in service

VPC

vpc-06f4de4de353cf931

Date created

January 1, 2025, 15:25 (UTC+05:30)

Scheme

Internet-facing

Hosted zone

Zone ID: ZP97RAFLXTNZK

Availability Zones

subnet-01e84f6728a2a04e9 ap-south-1a (aps1-az1)

subnet-0b29a28e814f3334b ap-south-1b (aps1-az3)

DNS name

a2caf46d4d2e34512b60a74b4968f9de-238144283.ap-south-1.elb.amazonaws.com (A Record)

This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. Learn more

Launch migration wizard

Distribution of targets by Availability Zone (AZ)

For each enabled Availability Zone, you can view the number of registered instances and their current health states. Selecting any values here will apply the corresponding filter to the Target instances table.

Listeners

Network mapping

Security

Health checks

Target instances

Monitoring

Attributes

Tags

Listeners

Manage listeners

A Classic Load Balancer listener uses the protocols and ports you configure, to both check for connection requests and forward received traffic to instances. The listener uses its protocol and port to check for connection requests. When traffic is received by the listener, it's forwarded to registered EC2 instances using the instance protocol and port.

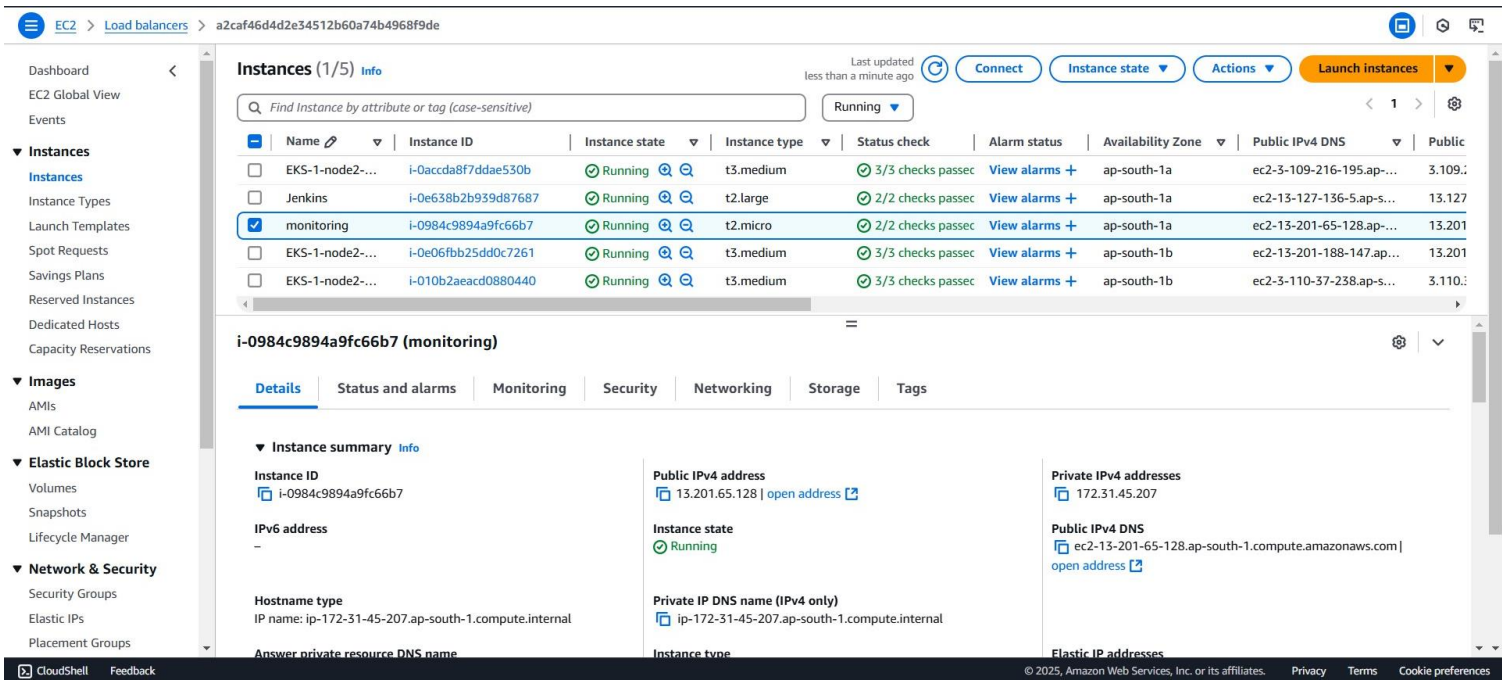
CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Monitoring Prometheus with Grafana

- 1. **Install Components:**  
Install Grafana, Prometheus, and Node Exporter on the monitoring server.
- 2. **Access Grafana:**  
Port: 3000, Username/Password: admin/admin.
- 3. **Connect Prometheus to Grafana:**  
Navigate to Data Sources → Add Prometheus → Enter Prometheus URL → Save & Test.
- 4. **Import Dashboard:**  
Click → Import → Enter Dashboard ID 1860 → Load → Select Prometheus → Import



Prometheus Alerts Graph Status Help

Use local time

Enable query history

Enable autocomplete

Enable highlighting

Enable linter

Expression (press Shift+Enter for newlines)

Execute

Table

Graph

Evaluation time

No data queried yet

Add Panel

Remove Panel

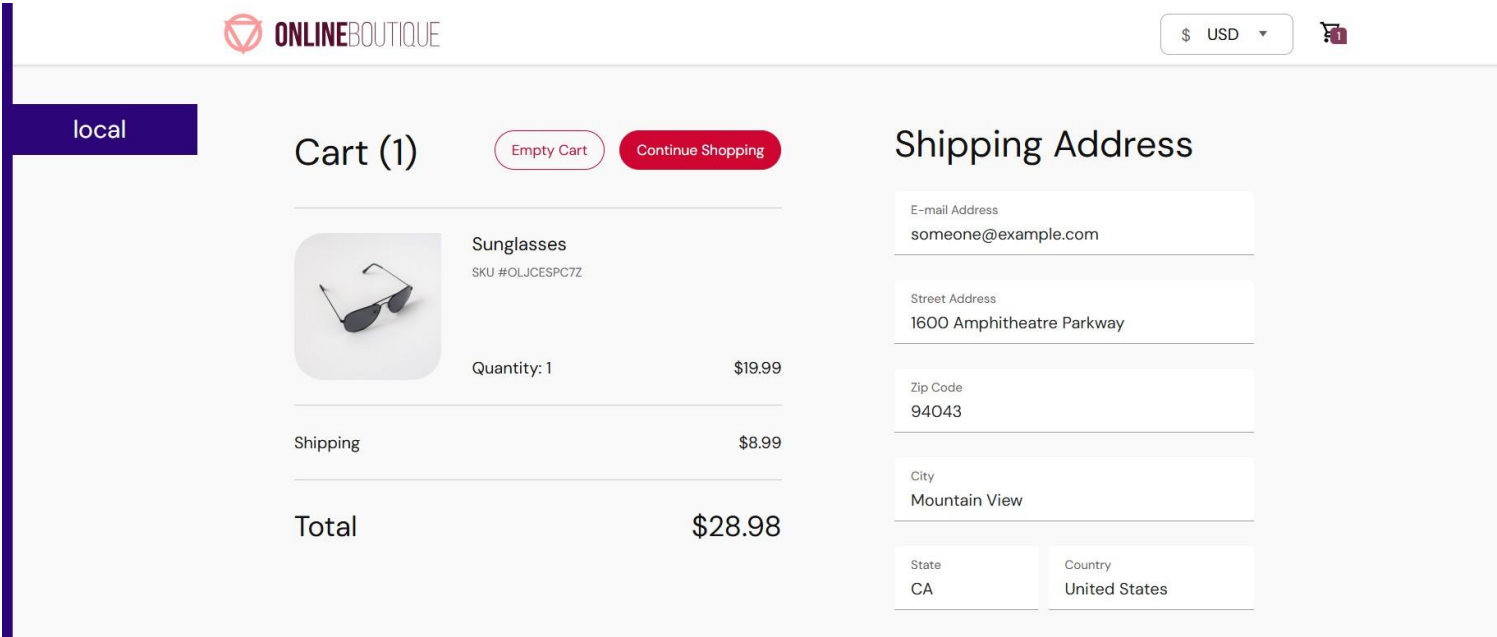
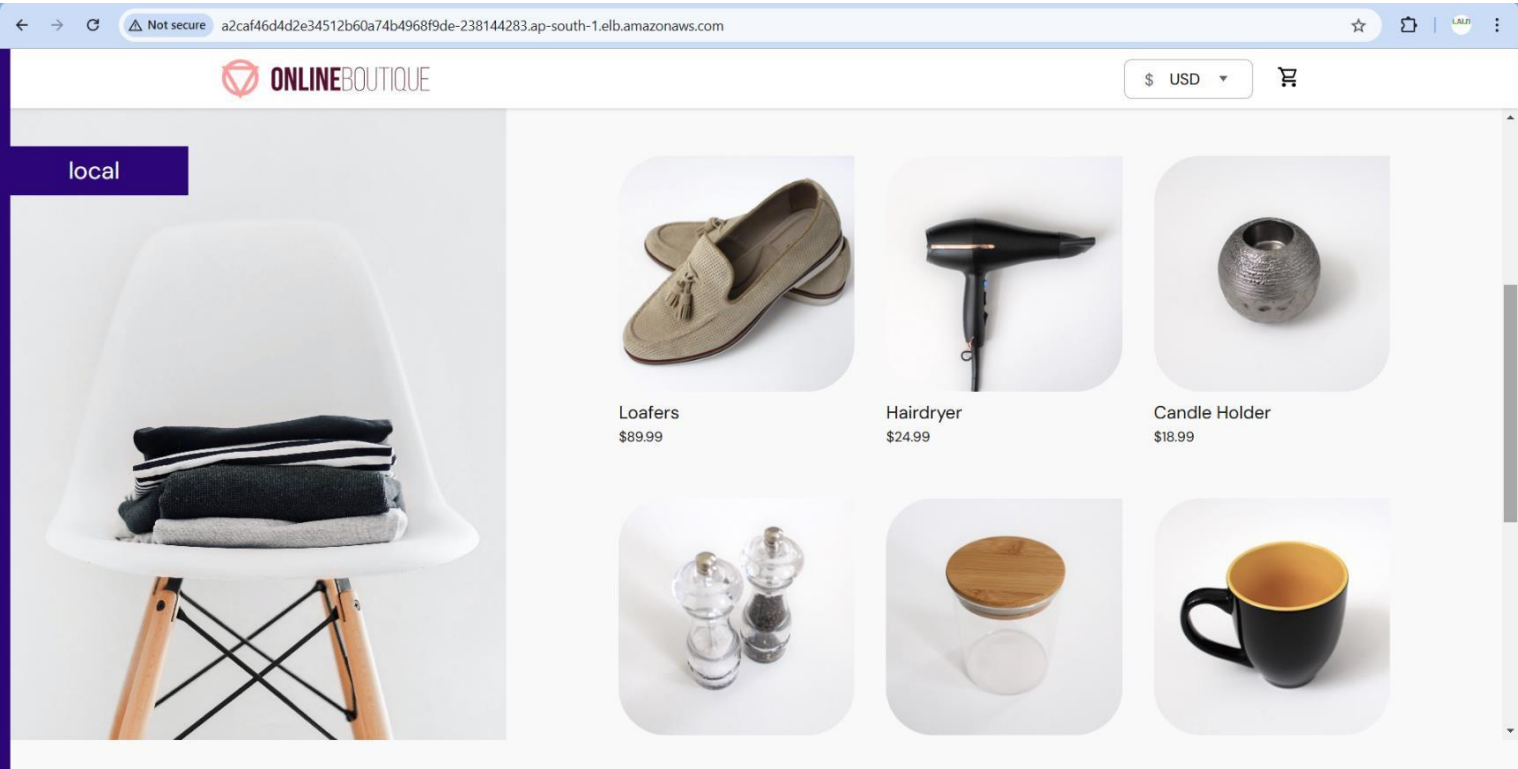
← → ↺ ⚠ Not secure 13.201.65.128:9100

# Node Exporter

[Metrics](#)

The screenshot displays the Prometheus Node Exporter Full dashboard. The interface includes a sidebar with navigation options like Home, Bookmarks, Starred, Dashboards, Explore, Alerting, Connections, and Administration. The main content area shows a 'Quick CPU / Mem / Disk' section with several gauge charts for Pressure, CPU Busy (5.3%), Sys Load (0.0%), RAM Used (36.0%), SWAP (N/A), and Root FS (16.0%). Below this is a 'Basic CPU / Mem / Net / Disk' section with four line graphs: CPU Basic, Memory Basic, Network Traffic Basic, and Disk Space Used Basic. The CPU Basic graph shows various states like Busy System, Busy User, Busy lowlat, Busy IRQs, Busy Other, and Idle. The Memory Basic graph shows RAM Total, RAM Used, RAM Cache + Buffer, RAM Free, and SWAP Used. The Network Traffic Basic graph shows traffic in Mb/s, and the Disk Space Used Basic graph shows disk usage percentage.

OUTPUT:





local



## Sunglasses

\$19.99

Add a modern touch to your outfits with these sleek aviator sunglasses.

1

Add To Cart

local

### Payment Method

Credit Card Number  
4432-8015-6152-0454

Month  
January

Year  
2026

CVV  
...

Place Order

### You May Also Like



Loafers



Hairdryer



Bamboo Glass Jar



Salt & Pepper Shakers

local

## Your order is complete!

We've sent you a confirmation email.

Confirmation # ff40efc4-c827-11ef-a3ff-3ed81f86d7f0

Tracking # XR-44532-229753957

Total Paid \$28.98

Continue Shopping

