

Step 1 – Deployment Strategy - (1)

Select one deployment tool (GitHub Actions / Codefresh / Harness / Spinnaker / etc.) and prepare a working pipeline for a simple service (ANY PBL OR BTECH PROJECT). Submit workflow file & pipeline diagram.

```
● (base) shruti@Shruti's-MacBook-Air devops % cd Rasoi-Genie
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % mkdir -p .github/workflows
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % touch .github/workflows/deploy.yml
```

```
Rasoi-Genie > .github > workflows > ! deploy.yml
 1   name: CI/CD Pipeline
 2
 3   on:
 4     push:
 5       branches:
 6         - main
 7
 8   jobs:
 9     build:
10       runs-on: ubuntu-latest
11
12     steps:
13       - name: Checkout Code
14         uses: actions/checkout@v3
15
16       - name: Set up Python
17         uses: actions/setup-python@v4
18         with:
19           python-version: '3.10'
20
21       - name: Install Backend Dependencies
22         run:
23           cd backend
24           pip install --upgrade pip
25           pip install -r requirements.txt
26
27       - name: Set up Node.js
28         uses: actions/setup-node@v3
29         with:
30           node-version: '20'
31
32       - name: Install Frontend Dependencies
33         run:
34           cd frontend
35           npm install --legacy-peer-deps
36
37       - name: Build Frontend
38         run:
39           cd frontend
40           npm run build
```

```

● (base) shruti@Shruti's-MacBook-Air devops % cd Rasoi-Genie
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % git add .github/workflows/deploy.yml
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % git commit -m "Add GitHub Actions workflow for frontend/backend CI"
[main a1e62cf] Add GitHub Actions workflow for frontend/backend CI
  1 file changed, 40 insertions(+)
   create mode 100644 .github/workflows/deploy.yml
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % git push origin main

Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 700 bytes | 700.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/SamidhaManjrekar/Rasoi-Genie.git
  4126bea..a1e62cf  main -> main

```

The screenshot shows the GitHub Actions pipeline for the repository 'SamidhaManjrekar / Rasoi-Genie'. The 'Actions' tab is selected. On the left, the 'CI/CD Pipeline' section is expanded, showing management options like Caches, Attestations, Runners, Usage metrics, and Performance metrics. The main area displays a single workflow run for 'deploy.yml'. The run is labeled '1 workflow run' and shows a green checkmark next to the commit message: 'Add GitHub Actions workflow for frontend/backend CI'. The commit hash is 'a1e62cf', it was pushed by 'shrutibist01' on the 'main' branch, and it happened 5 minutes ago. A timestamp of '45s' is also visible.

The screenshot shows the detailed view of the 'build' job from the previous workflow run. The 'Actions' tab is selected. The left sidebar shows the 'Summary' and 'Jobs' sections, with 'build' selected. The main area shows the job log with 14 steps: Set up job, Checkout Code, Set up Python, Install Backend Dependencies, Set up Node.js, Install Frontend Dependencies, Build Frontend, Post Set up Node.js, Post Set up Python, Post Checkout Code, and Complete job. Each step is marked with a green checkmark and a timestamp indicating its duration: 1s, 1s, 0s, 21s, 2s, 11s, 3s, 0s, 0s, 1s, and 0s respectively. A 'Re-run all jobs' button is located at the top right of the job log area.

Step 2 – Configuration Management & IaC - (2)

Use Ansible or Puppet to configure the runtime environment (install packages, create users, manage files). Upload your Playbook/Manifest & inventory.

```
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % mkdir ansible
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % cd ansible
● (base) shruti@Shruti's-MacBook-Air ansible % touch setup.yml inventory.ini
```

```
Rasoi-Genie > ansible > Ξ inventory.ini
1 [local]
2 localhost ansible_connection=local
```

```
Rasoi-Genie > ansible > ! setup.yml
1 ---
2   - hosts: all
3     connection: local
4     gather_facts: no
5
6     tasks:
7       - name: Ensure Homebrew is installed
8         command: /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
9         args:
10           creates: /opt/homebrew/bin/brew
11
12       - name: Install Python3 and Node using Homebrew
13         homebrew:
14           name: "{{ item }}"
15           state: present
16         loop:
17           - python
18           - node
19
20       - name: Create project directory
21         file:
22           path: ~/Desktop/college/devops/Rasoi-Genie
23           state: directory
24
25       - name: Create a venv folder for backend
26         command: python3 -m venv ~/Desktop/college/devops/Rasoi-Genie/backend/venv
27         args:
28           creates: ~/Desktop/college/devops/Rasoi-Genie/backend/venv
```

```
● (base) shruti@Shruti's-MacBook-Air ansible % ansible-playbook -i inventory.ini setup.yml

PLAY [all] ****
TASK [Ensure Homebrew is installed] ****
[WARNING]: Host 'localhost' is using the discovered Python interpreter at '/opt/homebrew/bin/python3.13', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.19/reference_appendices/interpreter_discovery.html for more information.
ok: [localhost]

TASK [Install Python3 and Node using Homebrew] ****
ok: [localhost] => (item=python)
ok: [localhost] => (item=node)

TASK [Create project directory] ****
ok: [localhost]

TASK [Create a venv folder for backend] ****
ok: [localhost]

PLAY RECAP ****
localhost                  : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Step 3 – Containerization & Orchestration - (1.5)

Dockerize your service, create Kubernetes manifests (Deployment, Service). Demonstrate rolling updates & rollback. Provide YAMLs & screenshots.

```
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % cd backend
● (base) shruti@Shruti's-MacBook-Air backend % touch Dockerfile
● (base) shruti@Shruti's-MacBook-Air backend % cd ..
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % cd frontend
● (base) shruti@Shruti's-MacBook-Air frontend % touch Dockerfile
● (base) shruti@Shruti's-MacBook-Air frontend % cd ..
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % mkdir k8s
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % cd k8s
● (base) shruti@Shruti's-MacBook-Air k8s % touch rasoi-deploy.yaml
```

Rasoi-Genie > frontend > 📄 Dockerfile

```
1  FROM node:20
2  WORKDIR /app
3  COPY package*.json ./
4  RUN npm install --legacy-peer-deps
5  COPY . .
6  RUN npm run build
7  # Serve static files
8  RUN npm install -g serve
9  EXPOSE 5173
10 CMD ["serve", "-s", "dist", "-l", "5173"]
```

Rasoi-Genie > backend > 📄 Dockerfile

```
1  FROM python:3.10-slim
2  WORKDIR /app
3  COPY . .
4  RUN pip install --no-cache-dir -r requirements.txt
5  EXPOSE 8000
6  CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

```
Rasoi-Genie > k8s > ! rasoi-deploy.yaml
 1  apiVersion: apps/v1
 2  kind: Deployment
 3  metadata:
 4    name: backend-deployment
 5  spec:
 6    replicas: 2
 7    selector:
 8      matchLabels:
 9        app: backend
10    template:
11      metadata:
12        labels:
13          app: backend
14      spec:
15        containers:
16          - name: backend
17            image: rasoi-backend:latest
18            ports:
19              - containerPort: 8000
20  ---
21  apiVersion: apps/v1
22  kind: Deployment
23  metadata:
24    name: frontend-deployment
25  spec:
26    replicas: 1
27    selector:
28      matchLabels:
29        app: frontend
30    template:
31      metadata:
32        labels:
33          app: frontend
34    spec:
35      containers:
36        - name: frontend
37          image: rasoi-frontend:latest
38          ports:
39            - containerPort: 5173
40  ---
41  apiVersion: v1
42  kind: Service
43  metadata:
44    name: backend-service
45  spec:
46    type: NodePort
47    selector:
48      app: backend
49    ports:
50      - port: 8000
51        targetPort: 8000
52        nodePort: 30008
53  ---
54  apiVersion: v1
55  kind: Service
56  metadata:
57    name: frontend-service
58  spec:
59    type: NodePort
60    selector:
61      app: frontend
62    ports:
63      - port: 5173
64        targetPort: 5173
65        nodePort: 30007
```

```
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % minikube delete
minikube start

🔥 Deleting "minikube" in docker ...
🔥 Deleting container "minikube" ...
Removing /Users/shruti/.minikube/machines/minikube ...
Removed all traces of the "minikube" cluster.
😊 minikube v1.37.0 on Darwin 15.7.1 (arm64)
⚡ Automatically selected the docker driver
🌐 Using Docker Desktop driver with root privileges
👍 Starting "minikube" primary control-plane node in "minikube" cluster
⚙️ Pulling base image v0.0.48 ...
💻 Downloading Kubernetes v1.34.0 preload ...
  > gcr.io/k8s-minikube/kicbase....: 450.06 MiB / 450.06 MiB 100.00% 6.61 Mi
  > preloaded-images-k8s-v18-v1....: 332.38 MiB / 332.38 MiB 100.00% 4.56 Mi
🔥 Creating docker container (CPUs=2, Memory=3072MB) ...
🌐 Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔧 Configuring bridge CNI (Container Networking Interface) ...
🌐 Verifying Kubernetes components...
  └─ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🌟 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % minikube status
kubectl cluster-info

minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

Kubernetes control plane is running at https://127.0.0.1:49952
CoreDNS is running at https://127.0.0.1:49952/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

```
(base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl apply -f k8s/rasoi-deploy.yaml
kubectl get pods -w

deployment.apps/backend-deployment created
deployment.apps/frontend-deployment created
service/backend-service created
service/frontend-service created
NAME                  READY   STATUS            RESTARTS   AGE
backend-deployment-54db9c55-j2wrbb   0/1    ContainerCreating   0          0s
backend-deployment-54db9c55-psgtb   0/1    ContainerCreating   0          0s
frontend-deployment-56d7d5f55f-dx972  0/1    ContainerCreating   0          0s
backend-deployment-54db9c55-psgtb   1/1    Running           0          1s
frontend-deployment-56d7d5f55f-dx972  1/1    Running           0          1s
backend-deployment-54db9c55-j2wrbb   1/1    Running           0          1s
```

Step 4 – Monitoring & Logging - (2)

Install Prometheus + Grafana (or Nagios). Expose app metrics & create a basic dashboard (uptime, latency, error rate). Upload dashboard screenshots.

```
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % minikube status

minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

@ (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl create namespace monitoring
Error from server (AlreadyExists): namespaces "monitoring" already exists
```

```

Rasoi-Genie > ! grafana-deploy.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: grafana
5    namespace: monitoring
6  spec:
7    replicas: 1
8    selector:
9      matchLabels:
10     app: grafana
11    template:
12      metadata:
13        labels:
14          app: grafana
15    spec:
16      containers:
17        - name: grafana
18          image: grafana/grafana:latest
19          ports:
20            - containerPort: 3000
21    ---
22  apiVersion: v1
23  kind: Service
24  metadata:
25    name: grafana-service
26    namespace: monitoring
27  spec:
28    type: NodePort
29    selector:
30      app: grafana
31    ports:
32      - port: 3000
33        targetPort: 3000
34        nodePort: 32000

```

```

● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl apply -f grafana-deploy.yaml
deployment.apps/grafana created
service/grafana-service created
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl get pods -n monitoring
NAME                  READY   STATUS            RESTARTS   AGE
grafana-567857c7d-6btvx  0/1    ContainerCreating   0          9s
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl get pods -n monitoring -w
NAME                  READY   STATUS    RESTARTS   AGE
grafana-567857c7d-6btvx  1/1    Running   0          63s

```

```
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl get svc -n monitoring
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
grafana-service	NodePort	10.96.203.151	<none>	3000:32000/TCP	3m23s

```
○ (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % minikube service grafana-service -n monitoring
```

NAMESPACE	NAME	TARGET PORT	URL
monitoring	grafana-service	3000	http://192.168.49.2:32000

✖ Starting tunnel for service grafana-service./	⋮	⋮	⋮	⋮	⋮
NAMESPACE	NAME	TARGET PORT	URL	⋮	⋮

monitoring	grafana-service	3000	http://192.168.49.2:32000	⋮	⋮
------------	-----------------	------	---------------------------	---	---

```
✖ Starting tunnel for service grafana-service.  
✖ Opening service monitoring/grafana-service in default browser...  
! Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```

```
Rasoi-Genie > ! prometheus-deployment-service.yaml
 1  apiVersion: apps/v1
 2  kind: Deployment
 3  metadata:
 4    name: prometheus-deployment
 5    namespace: monitoring
 6  spec:
 7    replicas: 1
 8    selector:
 9      matchLabels:
10        app: prometheus
11    template:
12      metadata:
13        labels:
14          app: prometheus
15      spec:
16        containers:
17          - name: prometheus
18            image: prom/prometheus:v2.43.0
19            ports:
20              - containerPort: 9090
21            args:
22              - "--config.file=/etc/prometheus/prometheus.yml"
23              - "--storage.tsdb.path=/prometheus/"
24            volumeMounts:
25              - name: prometheus-config
26                mountPath: /etc/prometheus/
27              - name: prometheus-data
28                mountPath: /prometheus/
29            volumes:
30              - name: prometheus-config
31                configMap:
32                  name: prometheus-config
33              - name: prometheus-data
34                emptyDir: {}
35  ---
36  apiVersion: v1
37  kind: Service
38  metadata:
39    name: prometheus-service
40    namespace: monitoring
41  spec:
42    type: NodePort
43    selector:
44      app: prometheus
45    ports:
46      - port: 9090
47        targetPort: 9090
48        nodePort: 32001
```

```
Rasoi-Genie > ! prometheus-namespace-config.yaml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: prometheus-config
5    namespace: monitoring
6  data:
7    prometheus.yml: |
8      global:
9        scrape_interval: 15s
10
11    scrape_configs:
12      # Scrape Prometheus itself
13      - job_name: "prometheus"
14        static_configs:
15          - targets: ["localhost:9090"]
16
17      # Scrape Node Exporter
18      - job_name: "node-exporter"
19        static_configs:
20          - targets: ["node-exporter.monitoring.svc.cluster.local:9100"]
21
```

```
Rasoi-Genie > ! node-exporter.yaml
```

```
1  apiVersion: apps/v1
2  kind: DaemonSet
3  metadata:
4    name: node-exporter
5    namespace: monitoring
6    labels:
7      app: node-exporter
8  spec:
9    selector:
10      matchLabels:
11        app: node-exporter
12    template:
13      metadata:
14        labels:
15          app: node-exporter
16      spec:
17        containers:
18          - name: node-exporter
19            image: prom/node-exporter:latest
20            ports:
21              - containerPort: 9100
22            volumeMounts:
23              - name: proc
24                mountPath: /host/proc
25                readOnly: true
26              - name: sys
27                mountPath: /host/sys
28                readOnly: true
29            securityContext:
30              runAsUser: 0
31        volumes:
32          - name: proc
33            hostPath:
34              path: /proc
35          - name: sys
36            hostPath:
37              path: /sys
38  ---
39  apiVersion: v1
40  kind: Service
41  metadata:
42    name: node-exporter
43    namespace: monitoring
44  spec:
45    selector:
46      app: node-exporter
47    ports:
48      - port: 9100
49        targetPort: 9100
```

```

● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl get pods -n monitoring -l app=node-exporter
No resources found in monitoring namespace.
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl apply -f node-exporter.yaml
daemonset.apps/node-exporter created
service/node-exporter created
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl get pods -n monitoring -l app=node-exporter
NAME          READY   STATUS      RESTARTS   AGE
node-exporter-h7g7m  0/1   ContainerCreating   0          7s
○ (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl port-forward svc/prometheus-service 9090:9090 -n monitoring
Forwarding from 127.0.0.1:9090 -> 9090
Forwarding from [::1]:9090 -> 9090
Handling connection for 9090
Handling connection for 9090
Handling connection for 9090

```

The screenshot shows the Prometheus Targets page. At the top, there are tabs for 'All scrape pools' (selected), 'All', 'Unhealthy', and 'Collapse All'. A search bar allows filtering by endpoint or labels. Below the tabs, there are two sections: 'node-exporter (1/1 up)' and 'prometheus (1/1 up)'. Each section contains a table with columns: Endpoint, State, Labels, Last Scrape, Scrape Duration, and Error.

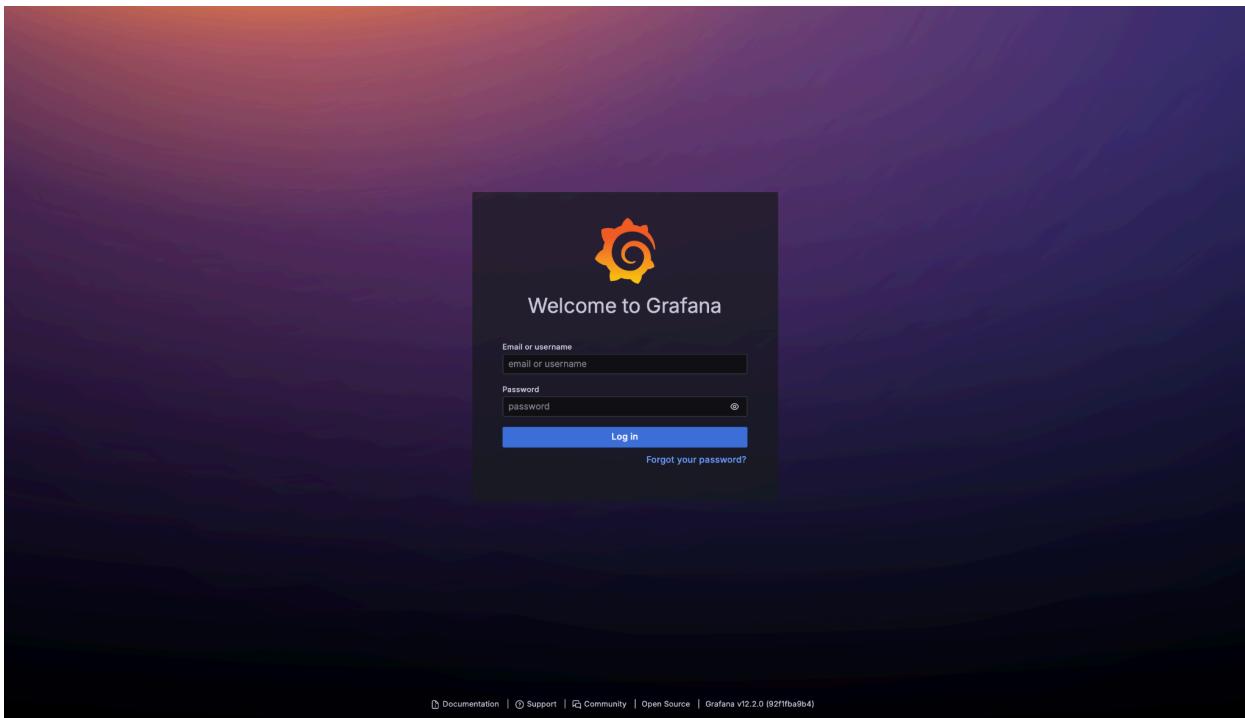
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://node-exporter.monitoring.svc.cluster.local:9100/metrics	UP	instances="node-exporter.monitoring.svc.cluster.local:9100" job="node-exporter"	14.382s ago	68.107ms	

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instances="localhost:9090" job="prometheus"	12.328s ago	11.264ms	

```

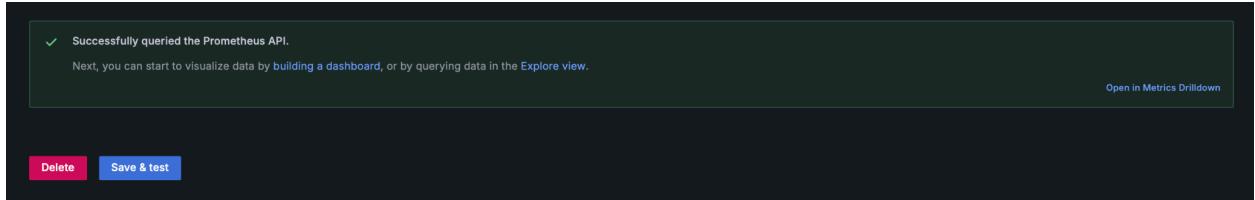
● (base) shruti@Shruti's-MacBook-Air Rasoi-Genie % kubectl get pods -n monitoring
NAME          READY   STATUS      RESTARTS   AGE
grafana-5f5d7cfdd5-xc864  1/1   Running   1 (10m ago)   16h
prometheus-deployment-55cf965cbb-78d86  1/1   Running   0          40s

```



The image shows the Grafana home dashboard. The left sidebar has a "Home" tab selected. The main content area has a title "Welcome to Grafana". It contains several panels: "Basic" (instructions for setup), "TUTORIAL DATA-SOURCE AND DASHBOARDS Grafana fundamentals" (with a "Learn how in the docs" link), "DATA SOURCES Add your first data source" (with a "Learn how in the docs" link), and "DASHBOARDS Create your first dashboard" (with a "Learn how in the docs" link). A "Need help?" section with links to Documentation, Tutorials, Community, and Public Slack is also present. The bottom left shows sections for "Dashboards", "Starred dashboards", and "Recently viewed dashboards". The bottom right shows a "Latest from the blog" section with a post about Taylor Swift.

The image shows the Grafana "Data sources" configuration page for Prometheus. The left sidebar has a "Data sources" tab selected. The main content area shows a "prometheus" entry with a "Type: Prometheus" label. It has tabs for "Settings" (selected) and "Dashboards". Under "Settings", there is a "Name" field set to "prometheus" with a "Default" toggle switch turned on. A note says "Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#)". A "Fields marked with * are required" note is also present. The bottom section is titled "Connection" with a "Prometheus server URL*" field containing "http://192.168.49.2:32001". On the right, there are buttons for "Type: Prometheus", "Alerting: Supported", "Explore data", and "Build a dashboard".



For this dashboard design used inbuilt dashboard design having code 1860.



Step 5 -Bonus

Participate in an external DevOps challenge (Kaggle/Devpost/Cloud hackathon). Attach proof of submission/leaderboard or GitHub PR badge.

kaggle

- + Create
- Home
- Competitions**
- Datasets
- Models
- Benchmarks
- Code
- Discussions
- Learn
- More

Your Work

- VIEWED
- Store Sales - Time S...
- Spaceship Titanic
- Natural Language Pr...
- NLP with Disaster tw...
- NLP-Disaster-Tweets-2

Search

Store Sales - Time Series Forecasting

Overview Data Code Models Discussion Leaderboard Rules Team Submissions

Submit Prediction ...

Rank	User	Profile	Score	Submissions	Time
406	Yosua Raffel		1.06641	1	18d
407	Sakshi Pathak		1.06992	1	8d
408	Samidha P		1.08776	1	1h
Your First Entry! Welcome to the leaderboard!					
409	Peterrr1		1.09807	2	2mo
410	Qi Chengxi		1.10029	4	6d
411	Andree Sendjaja		1.10486	1	10d
412	Everwalker2011		1.11070	1	1mo
413	Mayuna Mizutani		1.11362	7	6d
414	yuiop!		1.13737	7	1mo
-					

kaggle

- + Create
- Home
- Competitions**
- Datasets
- Models
- Benchmarks
- Code
- Discussions
- Learn
- More

Your Work

- VIEWED
- Store Sales - Time S...
- Spaceship Titanic
- Natural Language Pr...
- NLP with Disaster tw...
- NLP-Disaster-Tweets-2

Search

KAGGLE · GETTING STARTED PREDICTION COMPETITION - ONGOING

Store Sales - Time Series Forecasting

Use machine learning to predict grocery sales

Submit Prediction ...

Overview Data Code Models Discussion Leaderboard Rules Team Submissions

Recent ▾

Submissions

All Successful Errors

Submission and Description Public Score ⓘ

submission.csv
Complete · 1h ago **1.08776**