

## How to: Setup guide

1/ Log in to your [DigitalOcean](#) account. Navigate to the **Kubernetes** section and create a new Kubernetes cluster.

Search by resource name or public IP (Ctrl+B)

← Back to Clusters

## Create a Kubernetes cluster

### Select a version

We generally recommend the latest version unless your team has a specific need. Check out [DigitalOcean Kubernetes release notes](#).

1.32.2-do.0 - Recommended

### Choose a datacenter region

Your Kubernetes cluster will be located in a single datacenter.

Choose a datacenter region

Bangalore • Datacenter 1 • BLR1

**Note:** GPUs are supported in TOR1 and NYC2 datacenter regions

### VPC network

Nodes, pods, and services will belong to this [VPC network](#) and can communicate securely using their private IP addresses. Each datacenter has a default network.

default-blr1 DEFAULT

### Size pod and service networks ?

Network subnets can't be changed after creating the cluster.

Size network subnets for me RECOMMENDED  
We will assign non-overlapping subnets and size the network.

## 2/ Choose Cluster Capacity.

Select the Droplet, Node Plan and Min/Max nodes to configure cluster capacity settings.

## Choose cluster capacity ?

Select a plan that [best suits your workload type](#) ? for overall availability and performance. You can add or remove [nodes](#) and [node pools](#) at any time.

Node pool name

pool-lq06xtajw

Shared CPU

Dedicated CPU

Best for testing, low-traffic servers, and workloads that are not CPU-intensive. CPUs are shared between multiple other Droplets.

Machine type (Droplet) ?

Basic

Regular SSD



Node plan ?

**\$24/month per node (\$0.036/hour)**

4 GB total RAM / 2 vCPUs / 80 GB storage



☒ Set node pool to autoscale

Minimum nodes

—

1

+

to

Maximum nodes

—

3

+

Cost of all nodes: **\$24 - \$72/month** (\$0.04 - \$0.11/hour)


[Add Another Node Pool](#)

3/ Make your cluster more resilient and remove any single point of failure situation.


Check “**Add High Availability**” option, which will create replicas of kubernetes’s control plane components.

## Select additional options

You can add these cluster upgrades at any time. They can't be removed from the cluster after they are added.

 We recommend high availability to eliminate a single point of failure and increase peace of mind for production workloads.

### Get extra reliability for critical workloads

A [high availability control plane](#)  creates multiple replicas of control plane components to maximize cluster access and uptime with a 99.95% SLA.



☒ Add high availability

 **\$40/month** (\$0.06/hour)

### Automate database management

Create and automatically link managed databases to your cluster with the [operator](#) .



☐ Add database operator **BETA**

Free 

4/ Set name for cluster and Confirm pricing details. Click on “Create Cluster”

## Finalize

You can change the cluster's name, project, and tags at any time.

**Name\***

Can only contain lowercase alphanumeric characters and dashes.

k8s-1-32-2-do-0-blr1-1745234881124

**Project**





 first-project 

**Tags**

Use tags to organize and relate resources. They are not added as labels or taints to the cluster.  
Tags may contain letters, numbers, colons, dashes, and underscores.


 TAM... 

### Total Cost


	<b>1 to 3 Basic nodes</b> 4 GB total RAM / 2 vCPUs / 80 GB storage Autoscaling enabled	\$24.00 - \$72.00
	<b>HA control plane - 99.95% SLA</b>	\$40.00
	<b>Total cost</b> 	<b>\$64 - \$112/month</b> (\$0.10 - \$0.17/hour)

Create Cluster

5/ Download the Kubernetes Config file



tam-poc-1

in  [first-project](#) / BLR1 - 1.32.2-do.0

Overview


Resources


Insights


Marketplace

Settings

Check out these recommendations

 **Connect a container registry**  
[Create or connect a registry](#) to store and manage your images.

 **Add DigitalOcean Resources**  
[Add a Load Balancer](#) for traffic distribution or [Volumes](#) for storage.

 **Install**  
[Deploy to](#) capabilities

# Overview

CONFIGURATION


CPU	Memory	Disk
2 vCPUs	4 GB	80 GB


Download Config File

⬇

Remind me how to use this file to connect to the cluster

NODE POOL STATUS

 **1/1 Running**

 [pool-lq06xtajw](#)

[View resource details](#)

## 6/ Create Container registry

Search by resource name or public IP (Ctrl+B) Create ?

# Create a container registry

Store your Docker images in private repositories and easily integrate with any DigitalOcean Kubernetes Cluster.

## Choose a subscription plan

Subscription plan

<input checked="" type="radio"/> <b>Starter</b> 1 Registry / 1 Repository / 500 MB Storage	Free
<input type="radio"/> <b>Basic</b> 1 Registry / 5 Repositories / 5 GB Storage	\$5.00/month
<input type="radio"/> <b>Professional</b> 1 Registry / Unlimited Repositories / 100 GB Storage	\$20.00/month

Basic and Professional storage overages are billed at \$0.02/GB. The Starter plan is limited to 500 MB with no overages. [Learn more about pricing](#)

## Create a registry

Choose a datacenter region

**Bangalore** • Datacenter 1 • BLR1

Enter a unique name  
Can only contain lowercase alphanumeric characters and dashes.

sunny-do-tam-demo

This registry's name cannot be changed, so choose wisely my friend.

### Summary

**Subscription plan** Free

Plan: Starter  
Registries: 1  
Repositories: 1  
Storage: 500 MB

**Total cost** Free

Create Registry

## 7/ Authenticate with registry with doctl.

```
saryas@iamsaryal:~/DO/docker/app$ doctl registry login --never-expire
```

Logging Docker in to registry.digitalocean.com

8/ Integrate registry with K8S cluster. This will add a secret to all namespaces for chosen Kubernetes clusters. Allowing read-only access to pull images from this container registry.



## Getting Started

Hide

- ✓ Create a container registry
- ✓ Connecting to the registry
- 3 Integrating with Kubernetes
- 4 Next steps

### Integrating with Kubernetes

This will add a secret to all namespaces for chosen Kubernetes clusters. Allowing read-only access to [pull images from this container registry](#).

**Tip:** We recommend integrating with clusters that share the same region for the best performance.

☒ Integrate all clusters

1 / 1

Need more granular control over namespace access? [You can use Docker credentials to create a secret for specific namespaces](#).

Save & Continue

Skip

9/ Build and Push Docker Image, authenticate to container registry and push image to repository.

```
saryas@iamsaryal:~/DO/docker/app$ sudo docker build -t registry.digitalocean.com/sunny-do-tam-demo/saas-app:latest ./
[+] Building 2.3s (10/10) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 177B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:9aa5793609640ecea2f06451a0d6f379330880b413f954933289cf3b27a78567
0.0s
=> [internal] load build context
0.0s
=> => transferring context: 128B
0.0s
=> CACHED [2/5] WORKDIR /app
0.0s
=> CACHED [3/5] COPY requirements.txt .
0.0s
=> CACHED [4/5] RUN pip install --upgrade -r requirements.txt
0.0s
=> CACHED [5/5] COPY . .
0.0s
=> exporting to image
0.0s
=> => exporting layers
0.0s
=> => writing image sha256:8747df5c35e5c2870aaa08b8b0924b4243307ee7000f78b9cea700f519d55fd0
0.0s
=> => naming to registry.digitalocean.com/sunny-do-tam-demo/saas-app:latest
0.0s
```

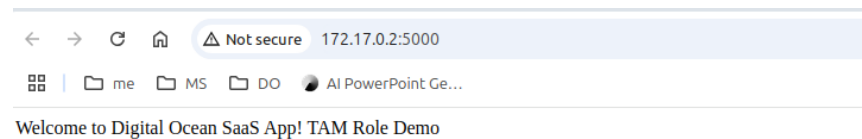
```
saryas@iamsaryal:~/D0/docker/app$ sudo docker login registry.digitalocean.com
Username: iamsaryal@gmail.com
Password:

WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
saryas@iamsaryal:~/D0/docker/app$ sudo docker tag saas-app:latest registry.digitalocean.com/sunny-do-tam-demo/saas-app:latest
Error response from daemon: No such image: saas-app:latest
saryas@iamsaryal:~/D0/docker/app$ sudo docker tag registry.digitalocean.com/sunny-do-tam-demo/saas-app:latest registry.digitalocean.com/sunny-do-tam-demo/saas-app:latest
saryas@iamsaryal:~/D0/docker/app$ sudo docker push registry.digitalocean.com/sunny-do-tam-demo/saas-app:latest
The push refers to repository [registry.digitalocean.com/sunny-do-tam-demo/saas-app]
c428e231e5e7: Pushed
afc96c6a8b4f: Pushed
32444e5ff190: Pushed
b84ec6a68e26: Pushed
7ad524989e4f: Pushed
e0cc49235ebd: Pushed
185439f46b41: Pushed
ea680fbff095: Pushed
latest: digest: sha256:366eb0043989915e96cfd45e74b9e09b500bfd5811966a3ae2ee49538d9dce64 size: 1990
saryas@iamsaryal:~/D0/docker/app$
```

10/ You can also test locally the image you've built.

```
saryas@iamsaryal:~/D0/docker/app$ sudo docker run registry.digitalocean.com/sunny-do-tam-demo/saas-app
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
172.17.0.1 - - [21/Apr/2025 13:53:01] "GET / HTTP/1.1" 200 -
```



11/ Verify cluster setup and

\$ Kubectl config view

```
saryas@iamsaryal:~/D0/docker/app$ k config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://2f9e97ca-fc9d-46dd-b3d9-dbbc1369324a.k8s.ondigitalocean.com
    name: do-blr1-tam-poc-1
contexts:
- context:
    cluster: do-blr1-tam-poc-1
    user: do-blr1-tam-poc-1-admin
    name: do-blr1-tam-poc-1
current-context: do-blr1-tam-poc-1
kind: Config
preferences: {}
users:
- name: do-blr1-tam-poc-1-admin
  user:
    token: REDACTED
saryas@iamsaryal:~/D0/docker/app$
```



12/ Deploy Kubernetes Resources:

\$ kubectl apply -f kubernetes/deployment.yaml

\$ kubectl apply -f kubernetes/service.yaml

\$ kubectl apply -f kubernetes/hpa.yaml

13/ Verify Deployment

\$ kubectl get pods,svc,hpa

```
saryas@iamsaryal:~/D0/docker/app$ kubectl get pods -o=wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE                                NOMINATED NODE   READINESS GATES
saas-app-867c8488b-694bn            1/1     Running   0           25m   10.108.0.47     pool-lq06xtajw-6hidu              <none>           <none>
saas-app-867c8488b-ncwmm            1/1     Running   0           25m   10.108.0.124    pool-lq06xtajw-6hidu              <none>           <none>

saryas@iamsaryal:~/D0/docker/app$ k get svc,ep
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/kubernetes                  ClusterIP           10.109.0.1       <none>            443/TCP           9h
service/saas-app-lb                 LoadBalancer       10.109.19.76     2400:6180:100:d0::7179:c001,68.183.246.102  80:32738/TCP     10m

NAME                                ENPOINTS                                AGE
endpoints/kubernetes                100.65.5.147:443                        9h
endpoints/saas-app-lb               10.108.0.124:5000,10.108.0.47:5000     10m
saryas@iamsaryal:~/D0/docker/app$
```

check service running fine inside pods.

root@saas-app-867c8488b-694bn:/app# curl localhost:5000

```
saryas@iamsaryal:~/D0/docker/app$ k exec -it pod/saas-app-867c8488b-694bn -- /bin/sh
# bash
root@saas-app-867c8488b-694bn:/app# apt-get install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.88.1-10+deb12u12).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
root@saas-app-867c8488b-694bn:/app# curl localhost:5000
Welcome to Digital Ocean SaaS App! TAM Role Demoroot@saas-app-867c8488b-694bn:/app#
root@saas-app-867c8488b-694bn:/app#
```

Testing using service/Loadbalancer public endpoint also (from CLI)

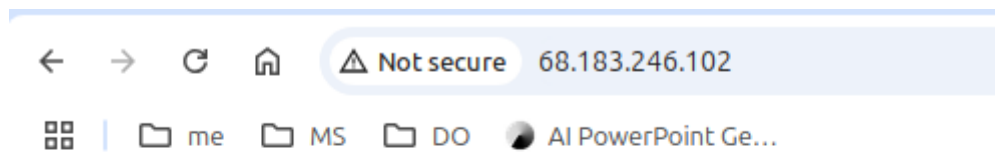
saryas@iamsaryal:~/DO/docker/app\$ curl 68.183.246.102 ### Loadbalancer external IP

**Welcome to Digital Ocean SaaS App! TAM Role Demo**

saryas@iamsaryal:~/DO/docker/app\$

```
saryas@iamsaryal:~/D0/docker/app$ kubectl get pods -o=wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE                                NOMINATED NODE   READINESS GATES
saas-app-867c8488b-694bn            1/1     Running   0           53m   10.108.0.47     pool-lq06xtajw-6hidu              <none>           <none>
saas-app-867c8488b-ncwmm            1/1     Running   0           53m   10.108.0.124    pool-lq06xtajw-6hidu              <none>           <none>
saryas@iamsaryal:~/D0/docker/app$ k get svc -o=wide
NAME                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE   SELECTOR
kubernetes          ClusterIP     10.109.0.1    <none>         443/TCP          9h    <none>
saas-app-lb         LoadBalancer 10.109.19.76  2400:6180:100:d0::7179:c001,68.183.246.102  80:32738/TCP    37m   app=saas-app
saryas@iamsaryal:~/D0/docker/app$
saryas@iamsaryal:~/D0/docker/app$
saryas@iamsaryal:~/D0/docker/app$
saryas@iamsaryal:~/D0/docker/app$ curl 68.183.246.102
Welcome to Digital Ocean SaaS App! TAM Role Demosaryas@iamsaryal:~/D0/docker/app$
```

## Access LB external IP - From browser:



Welcome to Digital Ocean SaaS App! TAM Role Demo

14/ Check and confirm that loadbalancing is working fine.

```
saryas@iamsaryal:~/DO/docker/app$ k get pods -o=wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE                                NOMINATED NODE   READINESS GATES
saas-app-867c8488b-694bn            1/1     Running   0           66m   10.108.0.47     pool-lq06xtajw-6hidu              <none>           <none>
saas-app-867c8488b-ncwmm            1/1     Running   0           66m   10.108.0.124    pool-lq06xtajw-6hidu              <none>           <none>
```

## ### POD 1

```
saryas@iamsaryal:~/DO/docker/app$ k logs --follow saas-app-867c8488b-694bn
```

\* Serving Flask app 'app'

\* Debug mode: off

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on all addresses (0.0.0.0)

\* Running on http://127.0.0.1:5000

\* Running on http://10.108.0.47:5000

Press CTRL+C to quit

10.122.0.3 - - [21/Apr/2025 21:31:08] "GET / HTTP/1.1" 200 -

10.122.0.3 - - [21/Apr/2025 21:31:23] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [21/Apr/2025 21:47:47] "GET / HTTP/1.1" 200 -

```
saryas@iamsaryal:~/D0/docker/app$ k logs --follow saas-app-867c8488b-694bn
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.108.0.47:5000
Press CTRL+C to quit
10.122.0.3 - - [21/Apr/2025 21:31:08] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:31:23] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/Apr/2025 21:47:47] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:48:30] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:50:33] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:50:43] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:50:47] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:53:03] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:53:04] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:53:05] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:53:05] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:53:38] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/Apr/2025 21:53:53] "GET / HTTP/1.1" 200 -
```

### ### POD 2

```
saryas@iamsaryal:~/D0/docker/app$ k logs --follow saas-app-867c8488b-ncwmm
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.108.0.124:5000
Press CTRL+C to quit
10.122.0.3 - - [21/Apr/2025 21:22:15] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:22:53] "GET / HTTP/1.1" 200 -
```

```
saryas@iamsaryal:~/D0/docker/app$ k logs --follow saas-app-867c8488b-ncwmm
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.108.0.124:5000
Press CTRL+C to quit
10.122.0.3 - - [21/Apr/2025 21:22:15] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:22:53] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:50:44] "GET /favicon.ico HTTP/1.1" 404 -
10.122.0.3 - - [21/Apr/2025 21:50:47] "GET / HTTP/1.1" 200 -
10.122.0.3 - - [21/Apr/2025 21:50:48] "GET / HTTP/1.1" 200 -
```

### 15/ Install Metrics Server (refer: [doc link](#))

```
saryas@iamsaryal:~/D0/docker$ helm install metrics-server metrics-server/metrics-server --version "$HELM_CHART_VERSION" --namespace metrics-server --create-namespace -f "Kubernetes-Starter-Kit-Developers/07-scaling-application-workloads/assets/manifests/metrics-server-values-v3.8.2.yaml"
NAME: metrics-server
LAST DEPLOYED: Tue Apr 22 04:32:35 2025
NAMESPACE: metrics-server
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
```

```
*****
* Metrics Server *
*****

Chart version: 3.12.2
App version: 0.7.2
Image tag: registry.k8s.io/metrics-server/metrics-server:v0.7.2
```

```
saryas@iamsaryal:~/D0/docker/app$ sudo apt-get install helm
[sudo] password for saryas:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package helm
saryas@iamsaryal:~/D0/docker/app$ cd ../
saryas@iamsaryal:~/D0/docker$ helm install metrics-server metrics-server/metrics-server --version "$HELM_CHART_VERSION" --namespace metrics-server --create-n
amespace -f "Kubernetes-Starter-Kit-Developers/07-scaling-application-workloads/assets/manifests/metrics-server-values-v3.8.2.yaml"
NAME: metrics-server
LAST DEPLOYED: Tue Apr 22 04:32:35 2025
NAMESPACE: metrics-server
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
*****
* Metrics Server *
*****
Chart version: 3.12.2
App version: 0.7.2
Image tag: registry.k8s.io/metrics-server/metrics-server:v0.7.2
*****
```

## 16/ Setup HorizontalPodAutoscaler (HPA)

```
saryas@iamsaryal:~/D0/docker/app$ kubectl autoscale deployment saas-app --cpu-percent=20 --min=1 --max=5
horizontalpodautoscaler.autoscaling/saas-app autoscaled
```

```
saryas@iamsaryal:~/D0/docker/app$ kubectl get hpa
NAME      REFERENCE          TARGETS      MINPODS  MAXPODS  REPLICAS  AGE
saas-app  Deployment/saas-app  cpu: 1%/20%  1        5        0         8s
```

```
saryas@iamsaryal:~/D0/docker$ k get hpa
NAME      REFERENCE          TARGETS      MINPODS  MAXPODS  REPLICAS  AGE
saas-app  Deployment/saas-app  cpu: 1%/20%  1        5        1         39m
```