CONFIGURATION GUIDES › KUBERNETES

# Local Kubernetes Monitoring

Applicare 10.0 · Updated March 24, 2026 · kubernetes · **Pending technical review**

**Step 1:** Run **kubectl top nodes** in your Kubernetes environment. If you do not see the details as expected, proceed to **Step 2** if the information is displayed correctly, move on to **Step 3**.

```
C:\Windows\system32>kubectl top nodes
NAME       CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
minikube   181m         9%     1601Mi          41%
```

**Step 2:** Install the metrics server in your Kubernetes cluster. Execute the following command to deploy the metrics pod.

**kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml**

Check if the metrics server is running by executing the following command.

**kubectl get pods -n kube-system**

```
E:\kubernetes>kubectl get pods -n kube-system
NAME                               READY   STATUS    RESTARTS        AGE
coredns-6d4b75cb6d-l7n9m           1/1     Running   0               6d21h
etcd-minikube                      1/1     Running   0               6d21h
kube-apiserver-minikube            1/1     Running   0               6d21h
kube-controller-manager-minikube   1/1     Running   1 (6d21h ago)   6d21h
kube-proxy-kmsnd                   1/1     Running   0               6d21h
kube-scheduler-minikube            1/1     Running   0               6d21h
metrics-server-658867cdb7-m46jm    1/1     Running   0               6d21h
storage-provisioner                1/1     Running   3 (2d4h ago)    6d21h

E:\kubernetes>
```

## If the metrics server is running, proceed to step 3. If it is not running, follow the steps below

**1:** Edit the configuration file by running the following command.

```bash
kubectl edit deployments.apps -n kube-system metrics-server
```

**2:** Add the following lines under in the **spec: container: args:** in the configuration file and save your changes.

```text
- --kubelet-insecure-tls
```

```
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        k8s-app: metrics-server
    spec:
      containers:
      - args:
        - --cert-dir=/tmp
        - --secure-port=4443
        - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
        - --kubelet-use-node-status-port
        - --metric-resolution=15s
        - --kubelet-insecure-tls
        image: k8s.gcr.io/metrics-server/metrics-server:v0.6.1
        imagePullPolicy: IfNotPresent
        livenessProbe:
          failureThreshold: 3
          httpGet:
```

**3:** After a few minutes, run the command **kubectl get pods -n kube-system** to check if the metrics server is running. If it is, running skip the following step.

**4:** Edit the configuration file by running the command **kubectl edit deployments.apps -n kube-system metrics-server** update the lines under **spec: container: args:** and save your changes.

```
TEXT
- --kubelet-preferred-address-types=InternalIP
```

```
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        k8s-app: metrics-server
    spec:
      containers:
      - args:
        - --cert-dir=/tmp
        - --secure-port=4443
        - --kubelet-preferred-address-types=InternalIP
        - --kubelet-use-node-status-port
        - --metric-resolution=15s
        - --kubelet-insecure-tls
        image: k8s.gcr.io/metrics-server/metrics-server:v0.6.1
        imagePullPolicy: IfNotPresent
        livenessProbe:
          failureThreshold: 3
          httpGet:
```

**5:** To verify that the metrics API server is running, execute the following command and review the output:

**kubectl top nodes**

```
C:\Windows\system32>kubectl top nodes
NAME       CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
minikube   181m         9%     1601Mi          41%
```

SingleAgent should be deployed in the machine where Kubernetes cluster is running.

**Step 3:** Refer here to **install SingleAgent and start.**

Once the agent is started, it will automatically be added to the Applicare Console.

**Step 4.** Run the following command and record the **cluster server** details for configuration.

```bash
kubectl config view
```

```
C:\Users\Anand>kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://kubernetes.docker.internal:6443
  name: docker-desktop
contexts:
- context:
    cluster: docker-desktop
    user: docker-desktop
  name: docker-desktop
current-context: docker-desktop
kind: Config
preferences: {}
users:
- name: docker-desktop
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
```

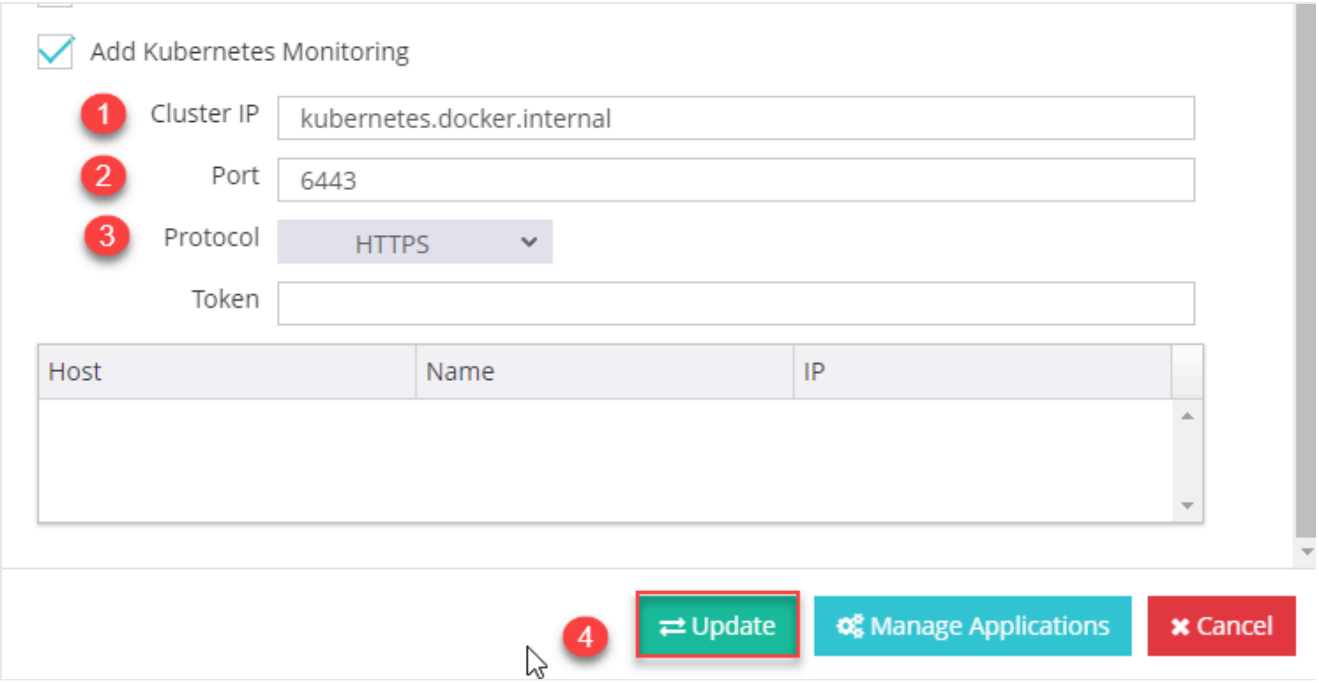
**Cluster IP :** kubernetes.docker.internal

**Port :** 6443

**Protocol :** https

**Step 5:** Log in to Applicare Console and navigate to the **Overview screen,** select the newly added server and click **Edit** button.

**Step 6: Enable** the **Add Kubernetes Monitoring checkbox** and enter the configuration details. Leave the **Token** field empty and click the **update** button.

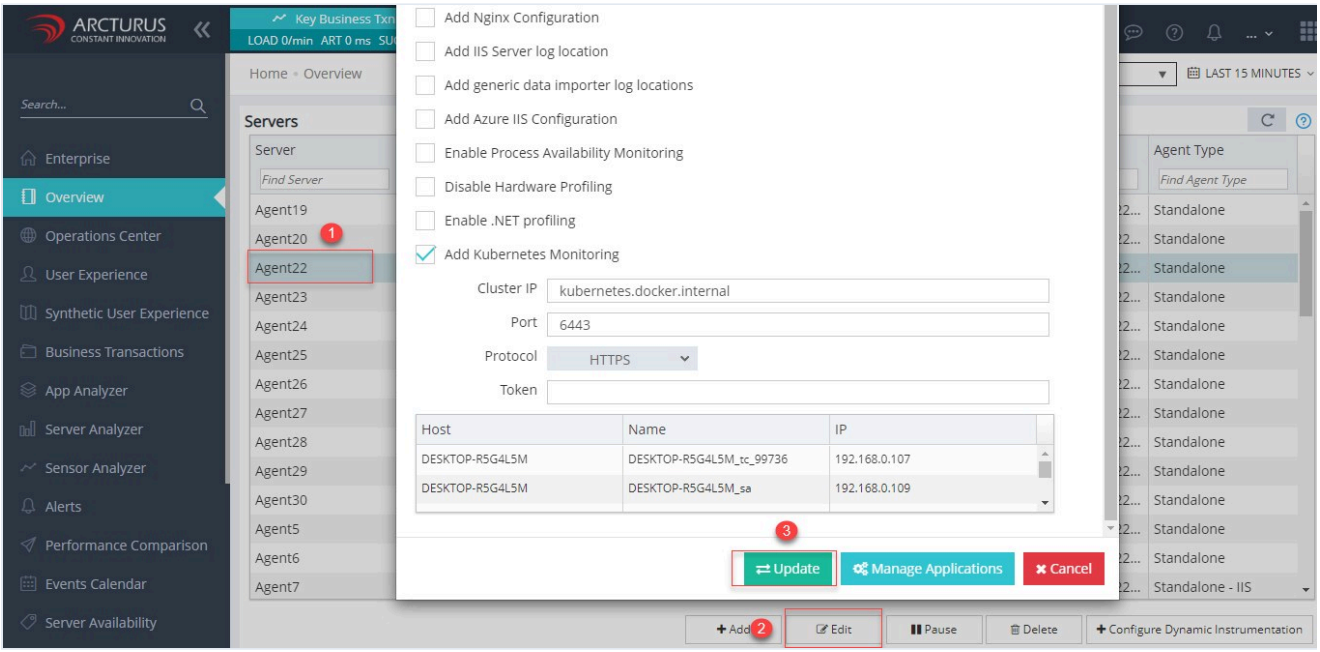



We can edit or delete the kubernetes configuration from both the **Overview** or **Kubernetes** menu.

## Overview Menu

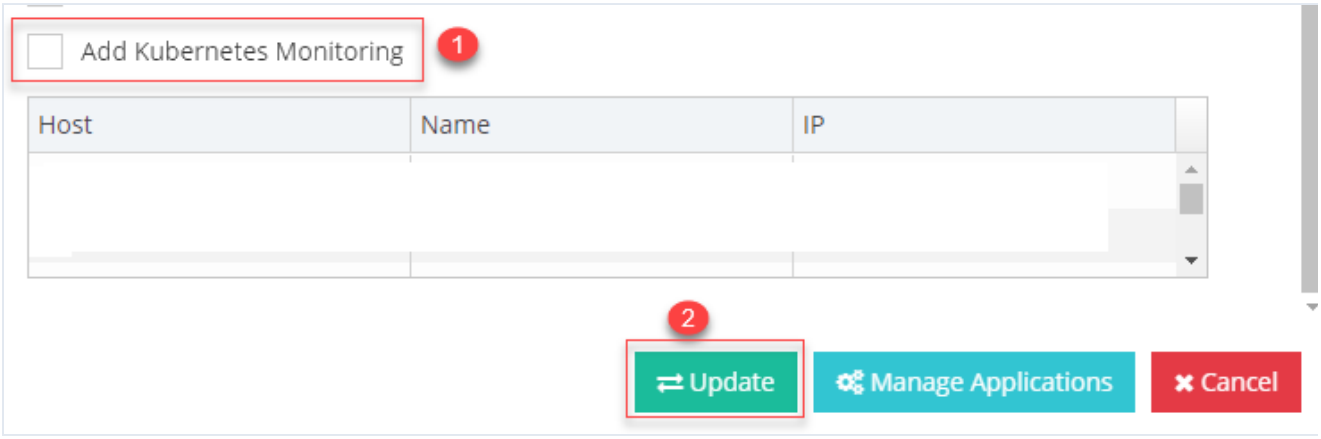
### Update

Select the **row** and click **Edit** button and update the details and click **Update** button.




## Delete

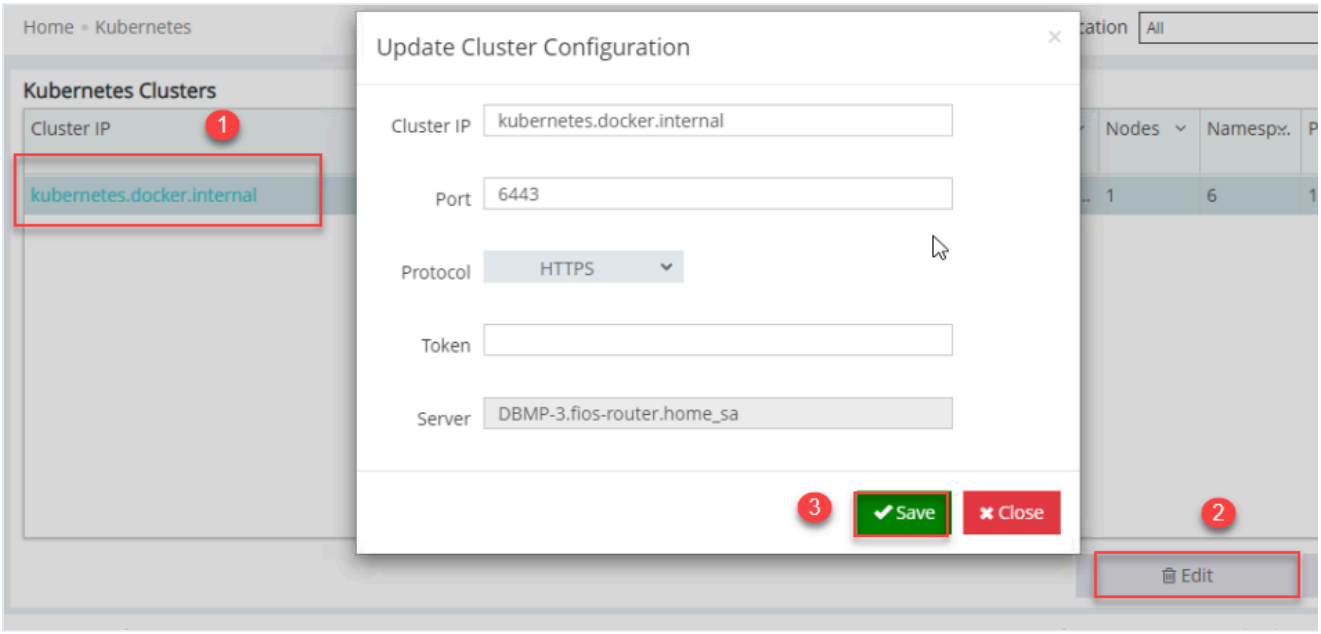
Select the **row** and click **Edit** button and uncheck the **Add the Kubernetes Monitoring** checkbox and click **Update** button.




## Kubernetes Menu

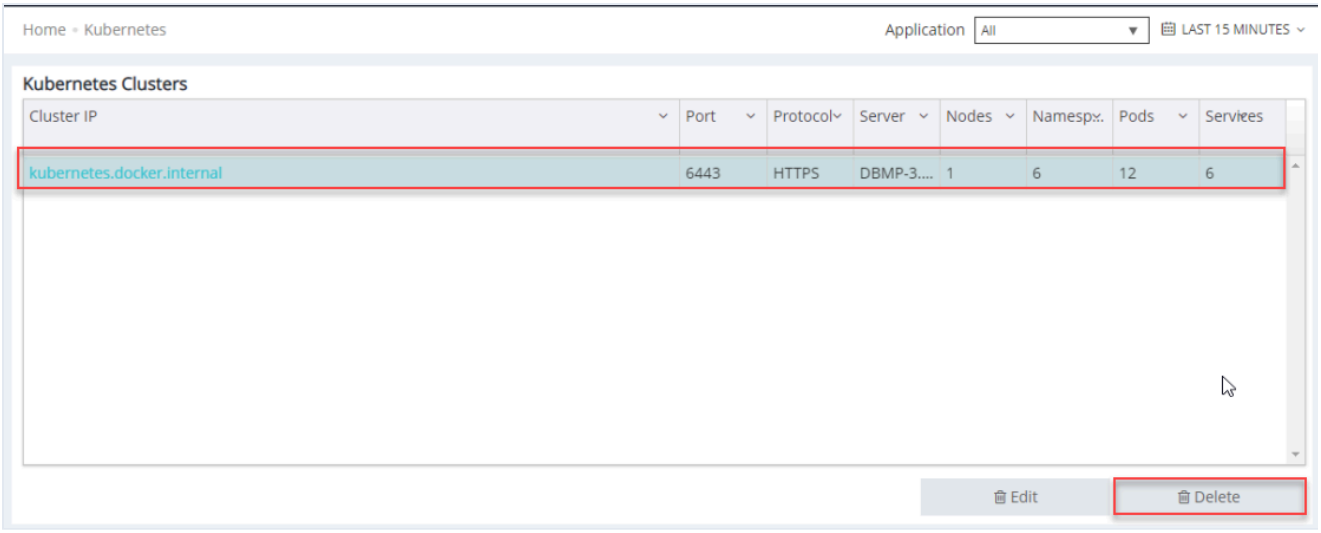
### Update

Select the **row** and click **Edit** button and update the details and click **Save** button.




### Delete

Select the **row** and click **Delete** button.


Home • Kubernetes
Application
All
LAST 15 MINUTES
Kubernetes Clusters
Cluster IP
Port
Protocol
Server
Nodes
Namespx.
Pods
Services
kubernetes.docker.internal
6443
HTTPS
DBMP-3....
1
6
12
6
Edit
Delete