

Installing Soffid on Kubernetes

Guide to show the installation process of Soffid IAM in Kubernetes

- [Installing IAM Console](#)
- [Installing Sync server](#)
- [How to copy Sync Server Kube Conf to Database table?](#)
- [How to copy to Kubernetes Secrets?](#)

Installing IAM Console

Guide to install IAM Console on Kubernetes.

Prerequisites

- Kubernetes
- 8GB RAM
- > 10GB disk space
- Supported database installed

Video Tutorial

Linux

<https://www.youtube.com/embed/kcMO1DZeD4w?rel=0>

Installation

You can use the docker image described at [Installing IAM console using Docker](#). Here you have a sample Kubernetes YAML descriptor to deploy it.

Mind that any certificate present in the folder `/opt/soffid/iam-console-3/trustedcerts` is considered as a trusted certificate. It is important to include the root syncserver certificate or any other certificate the console must connect with.

Another aspect to be aware of is the DNS resolution cache implemented by the java virtual machine. Because pods and service names often change its IP address, it suggested to disable the DNS cache adding the **-Dsun.net.inetaddr.ttl=-1** parameter.

MIIgCCCBVigAwIBAgIRAOFY+IkZ+FTddCqKixlQEIMwDQYJkoZIhvcNAQELBQAwwGy8xXzAjBgNVBAYTAkdCMRswGQYDVQQIEjHcmVhdGVyIE1hbmnNoZXN0ZXIxEDAQBGNVBACtTB1NhbGZvcmQxGDAWBGNVBAAOTD1NIY3RpZ28gTGltaXRIZDE3MDUGA1UEAxMuU2VjdGlnbyBSU0EgRG9tYWluIFZhbkGlkYXRpb24gU2VjdXJlIFNlcmlciBDQTAeFw0xOTA2MTgwMDAwMDBaFw0yMTA2MTcyMzU5NTlaMFkxITAFBgNVBAStGERvbWFPbiBDb250cm9sIFZhbkGlkYXRIZDEdMBsGA1UECxMUUG9zaXRpdmtVTU0wgV2lsZGNhcnQxFTATBgNVBAMMDCouc29mZmlkLmNvbTCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKnDKURLct1XfaMjmIU8QtXdhVe1XG1Oo4LrrEyUVBaAA/5RPcWrvcClf2Kq6/JTBBxbwvJP1pHANinwTGLam2INTL2jvlyYXC/oA0hqBRxDcbjkq7e7fj6R2rcFjCAx0jUiylzfGzmP/QX+ju7KrJ33sR4DPAG47Xnz8XgWJMuxDoSvQ8NeaWNAUjK7Pt3vHB/QD40MAAIsXuOq1w11R3MzeJvOnHgNPvxqGvVdHTDX5RwHoVEMEHF7I QY0Mh2olejQgN+vPOJNjh6vd7HiVUIVLXop8qhjjQgy2DQS2VGTUBObTFTgD81UPKzZgRzlziU3RWimZMVgHjzDn9MmzkzCAwEAaOCAvowggL2MB8GA1UdlwQYMBaAFI2MXSRURyrhd+mb+ZsF4bgBJWHHMB0GA1UdDgQWBbTPiYczqwQVFTogNEQydqg0WGGnwzAOBGNVHQ8BAf8EBAMCBaaAwDAYDVR0TAQH/BAIwADAdBgNVHSUEFjAUBggrBgEFBQCDAQYIKwYBBQUHAwlwSQYDVR0gBEIwQDA0BgSrBgEEAbIAQICBzAIMCMGCCSGAQUBwIBFhdodHRwczoV3NlY3RpZ28uY29tL0NQZAIBGZngQwBAgEwgYQGccSGAQUBwEBBHgwdJBPBggrBgEFBQcwAoZDaHR0cDovL2Nydc5zZWNOaWdvLmNvbS9TZWN0aWdvUINBRG9tYWluVmFsawRhGlvlBINIY3VyZVNlcmlckNBLMnydDAjBggrBgEFBQcwAYYXAHR0cDovL29jc3Auc2VjdGlnby5jb20wlmwYDVR0RBbBwwGoIMKi5zb2ZmaWQuY29tgppzb2ZmaWQuY29tMIIBfgYKKwYBBAHWeQIEAgSCAW4EggFqAWGAdgC72d+8H4pxtZOUI5eqkntHOFeVCqtS6BqQImQ2jh7RhQAAAWtpdk7pAAAEAwBHMEUCIQCYc83CoGLtckCrDeTAph3U/+XMqwkEPwqEgi9bu7xNBglgKachYG2OED40K9pd9byRWUjy+Bev+5tVen+I8JD48XoadQBEIGUusO7Or8RAB9io/IjA2uaCvtjLMbU/0zOWtbabQAAAAtPdk7+AAAEAwBGMEQCIFqju2Q/TTq48nkobC87nrfgE9FQmUp4PI98U90ygJAiBoFsiy0kz2ZDNz+BeAVjqAj7UsnrNlv8vwG3V7rh6kxgb3AG9Tdquwx8DEZ2JkApFEV/3cVHBHZAsEAKQaNsgiaN9kTAAABa2I2Tv4AAAQDAEGWRglhAMLmnVu4rduXSiaC5pfbk6uQsceV6zEx1fgNjQXNupDwaIEatCh5VG2IC6iWy0chA/Pfc5ejmIgBAmHbYLxs9uiOWwwwDQYJKoZIhvcNAQELBQADggEBAB

qZ8Stnzkk/abCQTMjOhNsSswSZZ74mszAGrd+emh7/VhLeJ29AaoMiCF5j0uphx/t9id5UmKbqwuapo9E1NuAVQqDO
V1N0wV4Awa2nEivbDcuDCTMX6VtOK3DnCN9yLMdD6GF9xcwzsgz5wKXu2Dxwt4vw05KIM+4Myy91sEpifa62+q
dzR/Vfbv6SqeL1lzTDyHMzEtBu/4jL189VeSkTVvdKGT1g6eAMHTX562z7jjgTH23c2zoICEj9YPd+KUbt6/OO+Pljsj0Me
TzO1QImj2syqCE/O4tYyHOHOdHJcrVSP951nCu0bkH6MBUhFvgk8a6rjl8tcnZCpsdcNU=

apiVersion: apps/v1

kind: Deployment

metadata:

name: soffid-console

labels:

app: soffid

type: console

spec:

replicas: 1

selector:

matchLabels:

app: soffid

type: console

template:

metadata:

labels:

app: soffid

type: console

spec:

containers:

- name: soffid-console

image: soffid/iam-console:3.0.0

imagePullPolicy: Always

resources:

limits:

memory: 4Gi

requests:

memory: 2Gi

volumeMounts:

- name: trusted-certs-volume

mountPath: /opt/soffid/iam-console-3/trustedcerts

ports:

- containerPort: 8080

env:

- name: DB_USER

```
      value: soffid
    - name: DB_PASSWORD
      value: Super5ecret
    - name: JAVA_OPT
      value: "-Xmx4048m -Dsun.net.inetaddr.ttl=1"
    - name: DB_URL
      value: jdbc:mariadb://mariadb-service:3306/soffid
  imagePullSecrets:
    - name: regcred
  volumes:
    - name: trusted-certs-volume
  secret:
    secretName: trusted-certs
---
apiVersion: v1
kind: Service
metadata:
  name: iam-console-service
spec:
  selector:
    app: soffid
    type: console
  type: loadBalancer
  ports:
    - name: web
      protocol: TCP
      port: 8080
      targetPort: 8080
```

Linux commands

Apply the YAML file with the defining Kubernetes resources

```
kubectl apply -f syncserver.yaml
```

Check deployments

```
kubectl get deployments
```

Check pods: you can check pods and their status

```
kubectl get pods
```

View the IAM console log

```
kubectl logs <your-pod-iamconsole-name>
```

When the console is created, the password for the user *admin* will be *changeit* and it will be valid for 24 hours.

Now you can connect to Soffid Console <http://<Node-Ip>:<publish-port>/soffid> The first thing you must do is to change the admin user password.

Next Step: [Installing Sync server](#)

Installing Sync server

Guide to install Sync server on Kubernetes.

Prerequisites

Soffid IAM sync server requires the following requirements:

- Supported database installed
- Soffid Console Installed

Video Tutorial

Linux

<https://www.youtube.com/embed/XZFMLQ00kAA?rel=0>

Installation

You can use the docker image described at [Installing Sync server using Docker](#). Here you have a sample Kubernetes YAML descriptor to deploy it.

```
# Secrets to store syncserver configuration
apiVersion: v1
kind: Secret
metadata:
  name: syncserver
type: Opaque
data:
```

config: c3Nva20=

Service account for sync server

apiVersion: v1

kind: ServiceAccount

metadata:

name: syncserver

Role to access the sync server

kind: ClusterRole

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: syncserver

rules:

- verbs:

- get

- update

apiGroups:

- ""

resources:

- deployments

- pods/attach

- secrets

- secrets/syncserver

kind: RoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: syncserver

namespace: default

subjects:

- kind: ServiceAccount

name: syncserver

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: syncserver

apiVersion: apps/v1

kind: Deployment


```
metadata:
  name: syncserver01
  labels:
    app: soffid
    type: syncserver
spec:
  replicas: 1
  selector:
    matchLabels:
      app: soffid
      type: syncserver
  template:
    metadata:
      labels:
        app: soffid
        type: syncserver
    spec:
      serviceAccountName: syncserver
      containers:
        - name: syncserver
          image: soffid/iam-sync:3.0.0
          ports:
            - containerPort: 760
              name: syncserver-port
          readinessProbe:
            initialDelaySeconds: 5
            failureThreshold: 1
            httpGet:
              path: /diag
              scheme: HTTPS
              port: 760
          livenessProbe:
            initialDelaySeconds: 5
            timeoutSeconds: 3
            failureThreshold: 3
            httpGet:
              path: /diag
              scheme: HTTPS
              port: 760
      env:
```

```
- name: DB_USER
  value: soffid
- name: DB_PASSWORD
  value: 5uper5ecret
- name: SOFFID_HOSTNAME
  value: syncserver01.cloud.soffid.com
- name: SOFFID_MAIN
  value: "yes"
- name: KUBERNETES_CONFIGURATION_SECRET
  value: "syncserver"
- name: DB_URL
  value: jdbc:mariadb://mariadb-service/soffid
```

apiVersion: v1

kind: Service

metadata:

name: syncserver

spec:

externalTrafficPolicy: Local

type: LoadBalancer

selector:

app: soffid

type: syncserver

ports:

- name: syncserver

protocol: TCP

port: 760

targetPort: 760

Linux commands

Apply the YAML file with the defining Kubernetes resources

```
kubectl apply -f syncserver.yaml
```

Check deployments

```
kubectl get deployments
```

Check pods: you can check pods and their status

```
kubectl get pods
```

View Sync server log

```
kubectl logs <your-pod-syncserver-name>
```

Now you can connect to the IAM console <http://<Node-Ip>:<publish-port>/soffid> and check if Console and Syncserver are connected.

How to copy Sync Server Kube Conf to Database table?

When you install soffid Sync server in kubernetes, a properties file is generated. If this file is not saved in a permanent storage, it could be lost during the Sync Server upgrade process.

Here you are the steps to copy your Kube config to a data base table

1.-

```
unset KUBERNETES_CONFIGURATION_SECRET
```

2.-

```
export DB_CONFIGURATION_TABLE=syncserver
```

3.-

```
java -cp "/opt/soffid/iam-sync/bin/bootstrap.jar:/opt/soffid/iam-sync/lib/mariadb-java-client-1.8.0.jar:/opt/soffid/iam-sync/lib/ojdbc10-19.18.0.0.jar:/opt/soffid/iam-sync/lib/postgresql-42.2.5.jar:/opt/soffid/iam-sync/lib/sqljdbc4-3.0.jar" com.soffid.iam.sync.bootstrap.KubernetesSaver
```


How to copy to Kubernetes Secrets?

When making any manual changes to the Sync server configuration files, it will be necessary to copy these changes to the Kubernetes secrets.

Command example:

```
java -cp "/opt/soffid/iam-sync/bin/bootstrap.jar" com.soffid.iam.sync.bootstrap.KubernetesSaver
```

Soffid version 3.x upgrade automatically the certificates when the certificate end date is close and no manual actions are required.