Installation methods-Kubernetes via Helm Chart

Prerequisites

- You have extensive understanding of Kubernetes
- Installed Helm package manager version v3.11.3 or greater
- You have kubectl installed and connected to your kubernetes cluster and must have storageclas installed.

Install Infisical Helm repository:

```
helm repo add infisical-helm-charts 'https://dl.cloudsmith.io/public/infisical/helm-charts/helm/charts/'
helm repo update
helm pull infisical-helm-charts/infisical-standalone --untar
```

Now we need to update the values.yaml inside `infisical-standalone` folder.

```
# -- Overrides the default release name
nameOverride: ""

# -- Overrides the full name of the release, affecting resource names
fullnameOverride: ""

infisical:
  # -- Enable Infisical chart deployment
  enabled: true
  # -- Sets the name of the deployment within this chart
  name: infisical

  autoBootstrap:
    # -- Enable auto-bootstrap of the Infisical instance
    enabled: false
```

```
    image:
      # -- Infisical Infisical CLI image tag version
      tag: "0.41.86"

    # -- Template for the data/stringData section of the Kubernetes secret. Available
functions: encodeBase64
    secretTemplate: '{"data":{"token":"{{.Identity.Credentials.Token}}"}}'

    secretDestination:
      # -- Name of the bootstrap secret to create in the Kubernetes cluster which will store
the formatted root identity credentials
      name: "infisical-bootstrap-secret"

      # -- Namespace to create the bootstrap secret in. If not provided, the secret will be
created in the same namespace as the release.
      namespace: "default"

    # -- Infisical organization to create in the Infisical instance during auto-bootstrap
    organization: "default-org"

    credentialSecret:
      # -- Name of the Kubernetes secret containing the credentials for the auto-bootstrap
workflow
      name: "infisical-bootstrap-credentials"

  databaseSchemaMigrationJob:
    image:
      # -- Image repository for migration wait job
      repository: ghcr.io/groundnuty/k8s-wait-for
      # -- Image tag version
      tag: no-root-v2.0
      # -- Pulls image only if not present on the node
      pullPolicy: IfNotPresent

  serviceAccount:
    # -- Creates a new service account if true, with necessary permissions for this chart. If
false and `serviceAccount.name` is not defined, the chart will attempt to use the Default
service account
    create: true
    # -- Custom annotations for the auto-created service account
```

```
    annotations: {}
    # -- Optional custom service account name, if existing service account is used
    name: null

  # -- Override for the full name of Infisical resources in this deployment
  fullnameOverride: ""
  # -- Custom annotations for Infisical pods
  podAnnotations: {}
  # -- Custom annotations for Infisical deployment
  deploymentAnnotations: {}
  # -- Number of pod replicas for high availability
  replicaCount: 2

  image:
    # -- Image repository for the Infisical service
    repository: infisical/infisical
    # -- Specific version tag of the Infisical image. View the latest version here
https://hub.docker.com/r/infisical/infisical
    tag: "v0.151.0"
    # -- Pulls image only if not already present on the node
    pullPolicy: IfNotPresent
    # -- Secret references for pulling the image, if needed
    imagePullSecrets: []

  # -- Node affinity settings for pod placement
  affinity: {}
  # -- Tolerations definitions
  tolerations: []
  # -- Node selector for pod placement
  nodeSelector: {}
  # -- Topology spread constraints for multi-zone deployments
  # -- Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-
constraints/
  topologySpreadConstraints: []

  # -- Kubernetes Secret reference containing Infisical root credentials
  kubeSecretRef: "infisical-secrets"

  service:
    # -- Custom annotations for Infisical service
```

```
    annotations: {}
    # -- Service type, can be changed based on exposure needs (e.g., LoadBalancer)
    type: NodePort
    # -- Optional node port for service when using NodePort type
    nodePort: "30880"

  resources:
    limits:
      # -- Memory limit for Infisical container
      memory: 1000Mi
    requests:
      # -- CPU request for Infisical container
      cpu: 350m

ingress:
  # -- Enable or disable ingress configuration
  enabled: false
  # -- Hostname for ingress access, e.g., app.example.com
  hostName: ""
  # -- Specifies the ingress class, useful for multi-ingress setups
  ingressClassName: nginx

  nginx:
    # -- Enable NGINX-specific settings, if using NGINX ingress controller
    enabled: false

  # -- Custom annotations for ingress resource
  annotations: {}
  # -- TLS settings for HTTPS access
  tls: []
    # -- TLS secret name for HTTPS
    # - secretName: letsencrypt-prod
    # -- Domain name to associate with the TLS certificate
    #   hosts:
    #     - some.domain.com

postgresql:
  # -- Enables an in-cluster PostgreSQL deployment. To achieve HA for Postgres, we recommend
deploying https://github.com/zalando/postgres-operator instead.
  enabled: true
```

```
  # -- PostgreSQL resource name
  name: "postgresql"
  # -- Full name override for PostgreSQL resources
  fullnameOverride: "postgresql"

  image:
    # -- Image registry for PostgreSQL
    registry: mirror.gcr.io
    # -- Image repository for PostgreSQL
    repository: bitnamilegacy/postgresql

  auth:
    # -- Database username for PostgreSQL
    username: infisical
    # -- Password for PostgreSQL database access
    password: root
    # -- Database name for Infisical
    database: infisicalDB

  useExistingPostgresSecret:
    # -- Set to true if using an existing Kubernetes secret that contains PostgreSQL
connection string
    enabled: false
    existingConnectionStringSecret:
      # -- Kubernetes secret name containing the PostgreSQL connection string
      name: ""
      # -- Key name in the Kubernetes secret that holds the connection string
      key: ""

redis:
  # -- Enables an in-cluster Redis deployment
  enabled: true
  # -- Redis resource name
  name: "redis"
  # -- Full name override for Redis resources
  fullnameOverride: "redis"

  image:
    # -- Image registry for Redis
    registry: mirror.gcr.io
```

```
    # -- Image repository for Redis
    repository: bitnamilegacy/redis

  cluster:
    # -- Clustered Redis deployment
    enabled: false

  # -- Requires a password for Redis authentication
  usePassword: true

  auth:
    # -- Redis password
    password: "mysecretpassword"

  # -- Redis deployment type (e.g., standalone or cluster)
  architecture: standalone
```

FYI: I just disabled ingress and put a fixed nodeport

Now in the values.yaml as we can

```
# -- Kubernetes Secret reference containing Infisical root credentials
 kubeSecretRef: "infisical-secrets"
```

So we need create the secret called `infisical-secrets`

```
apiVersion: v1
kind: Secret
metadata:
  name: infisical-secrets
  namespace: default
type: Opaque
stringData:
  ENCRYPTION_KEY: "8a9cfb7128e349f27ef2e6d4a6b5cc93"
  AUTH_SECRET: "d5f47e8b4a23b9c1ef7f03cc69ad41e2"
  DB_CONNECTION_URI:
"postgresql://infisical:root@postgresql.default.svc.cluster.local:5432/infisicalDB"
  REDIS_URL: "redis://:mysecretpassword@redis-master.default.svc.cluster.local:6379"
```

```
  SITE_URL: "http://192.168.88.14:30880"
```

Now we need to apply the secret manifest first, then need to proceed with infisical helm chart installation.

Once the secret created, then we need to got the downloaded chart directory and need to run bellow command.

```
helm install infisical . -f values.yaml
```

Boom our infisical server now running on http://192.168.88.14:30880

https://infisical.com/docs/self-hosting/deployment-options/kubernetes-helm

---

Revision #7
Created 1 November 2025 14:07:38 by Admin
Updated 12 November 2025 12:59:33 by Admin