BioTuring System - GPU enterprise version installation guide

This edition of the installation guide describes the installation process of BioTuring® System for K8S.

1. Introduction

BioTuring System is a GPU-accelerated single-cell and spatial platform developed by BioTuring®. It dramatically increases the computing performance of single-cell and spatial analysis by harnessing the power of the graphics processing unit (GPU).

1.1. Pre-Installation Requirements

Before installing the BioTuring System on Linux/K8S, some pre-installation steps are required: - System: K8s - Each node has one or multiple NVIDIA GPU(s) (at least 16 GB memory per GPU) - SSL certificate and a domain name for users to securely access the platform on the web browser - A token obtained from BioTuring - At least 64 GB of root partition. - At least 32 GB of RAM - At least 16 CPU cores.

1.2. Self-Signed CA Certificate installation (Optional, just in case your node has a problem with curl https):

Adding self-signed certificates as trusted to your proxy agent/server

```
bash ./cert/install.sh
```

2. Prepare GPU toolkit for K8S

1. Patch container engines (Docker, Containerd)

Install NVidia container toolkit on each node following the guide: https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html

Check container engines (Docker, Containerd)

```
For microk8s :
microk8s kubectl describe no | grep Runtime

For vanilla :
kubectl describe no | grep Runtime
```

If container engine is Containerd, add these lines to : /etc/containerd/config.toml

```
privileged_without_host_devices = false
base_runtime_spec = ""
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
```

```
    SystemdCgroup = true
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.nvidia]
    privileged_without_host_devices = false
    runtime_engine = ""
    runtime_root = ""
    runtime_type = "io.containerd.runc.v1"
    [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.nvidia.options]
    BinaryName = "/usr/bin/nvidia-container-runtime"
    SystemdCgroup = true
[plugins."io.containerd.grpc.v1.cri".cni]
bin_dir = "/opt/cni/bin"
conf_dir = "/etc/cni/net.d"
```

After that, restart containerd

```
sudo systemctl restart containerd
sudo nvidia-container-cli --load-kmods info
```

If container engine is Docker, add these lines to : /etc/docker/daemon.json

```
{
    "default-runtime": "nvidia",
    "runtimes": {
        "nvidia": {
            "path": "nvidia-container-runtime",
            "runtimeArgs": []
        }
    }
}
```

After that, restart docker

```
sudo systemctl restart docker
sudo nvidia-container-cli --load-kmods info
```

3. Install BioTuring ecosystem on K8S

We support all k8s engines: GKE (Google Kubernetes Engine), EKS (Amazon Elastic Kubernetes Service), AKS (Azure Kubernetes Service), MicroK8s, and vanilla K8S.

1. Ensure that helm (version 3) is installed.

First, check the Helm version

```
Example :

microk8s enable helm3

microk8s helm3 version
```

2. Add BioTuring Helm charts

For Kubernetes

```
Example:

For Vanilla K8s:

helm repo add bioturing https://bioturing.github.io/charts/apps/

For Microk8s:

microk8s helm3 repo add bioturing https://bioturing.github.io/charts/apps/
```

3. Simple Installation (Recommended):

```
bash ./install.k8s.sh
```

Going through this interactive installation to finish the installation. After this step, just access the BioTuring System via the specified domain in the installation process. If it's not in the DNS, please add the ip/domain to the local machine DNS host file.

4. Check pods information

```
microk8s kubectl get all

microk8s kubectl get pods
microk8s kubectl get services --all-namespaces
microk8s kubectl get services
microk8s kubectl get pvc

microk8s kubectl logs bioturing-ecosystem-0

microk8s.kubectl -n ingress get pods
microk8s.kubectl -n ingress logs <your pod name here> | grep reload
```

5. Check secrets

- bioturing-ecosystem-tls
- bioturing-ecosystem
- bioturingregred

```
microk8s kubectl edit secrets mysecret

Example:
microk8s kubectl edit secrets bioturing-ecosystem-tls
```

6. Helm chart Values

Kubernetes: `>=1.19.0-0`

Key	Type	Default	Description
image.tag	string	`"1.0.21"`	image tag
secret.data.domain	string	`"bbrowserx.com"`	your domain
secret.data.ssodomains	string	`""`	allow domains
secret.data.bbtoken	string	`""`	bioturing access token
secret.admin.username	string	`admin`	username
secret.admin.password	string	`turing2022`	password
secret.server.useletsencrypt	string	`"false"`	
secret.server.lcall	string	`"C.UTF-8"`	
secret.server.lclang	string	`"C.UTF-8"`	
secret.server.certificate	string	`""`	CRT base64 string
secret.server.key	string	`""`	KEY base64 string
service.type	string	`ClusterIP`	
service.ports.http.port	int	80	
service.ports.https.port	int	443	
persistence.dirs.app.size	string	`5Gi`	APP size
persistence.dirs.app.storageClass	string	`""`	
persistence.dirs.user.size	string	`5Gi`	USER size
persistence.dirs.user.storageClass	string	`""`	
persistence.dirs.user.existingClaim	bool	`true`	
ingress.enabled	bool	`true`	
ingress.className	string	`""`	
ingress.annotations	object	`{}`	
ingress.tls.enabled	bool	`true`	
resources	object	`{}`	
autoscaling	object	`{}`	
nodeSelector	object	`{}`	
tolerations	object	`{}`	
affinity	object	`{}`	
podAnnotations	object	`{}`	
podSecurityContext	object	`{}`	
securityContext	object	`{}`	
serviceAccount.name	string	`""`	
gpu.enabled	bool	`true`	
gpu.runtimeClassName	string	`"nvidia"`	

For Containerd runtime :

```
gpu.runtimeClassName="nvidia"
```

For Docker runtime :

```
gpu.runtimeClassName=""
```

7. Manual Installation

Please replace paths to your certificate, key, admin password, and other helm chart values of your choice.

```
BBTOKEN="USE TOKEN OBTAINED FROM BIOTURING"
SSLCRT="base64 -w 0 ./bioturing.com.crt" # <- (REPLACE THIS WITH A PATH TO YOUR CRT CERTFICA
SSLKEY="base64 -w 0 ./bioturing.com.key" # <- (REPLACE THIS WITH A PATH TO YOUR KEY)
ADMIN_USERNAME="admin"
ADMIN_PASSWORD="admin" # <- (CHANGE YOUR PASSWORD IF NECESSARY)
USELETSENCRYPT="false"
SVHOST="k8stest.bioturing.com" # <- (CHANGE THIS TO YOUR K8S INGRESS DOMAIN)
CHART_VERSION="1.0.20" # <- (CHANGE IT IF NECESSARY)
LC_ALL="C.UTF-8" # <- (CHANGE IT IF NECESSARY)
LC_LANG="C.UTF-8" # <- (CHANGE IT IF NECESSARY)

For Microk8s:

microk8s helm3 repo update
microk8s helm3 registry login -u admin registry.bioturing.com
microk8s helm3 upgrade --install --set secret.data.bbtoken="${BBTOKEN}" \
 --set secret.data.domain="${SVHOST}" \
 --set secret.server.certificate="${SSLCRT}" \
 --set secret.server.key="${SSLKEY}" \
 --set secret.server.useletsencrypt="${USELETSENCRYPT}" \
 --set secret.server.lcall="${LC_ALL}" \
 --set secret.server.lclang="${LC_LANG}" \
 --set secret.admin.username="${ADMIN_USERNAME}" \
 --set secret.admin.password="${ADMIN_PASSWORD}" \
bioturing bioturing/ecosystem --version ${CHART_VERSION}

For Vanilla k8s:

helm repo update
helm registry login -u admin registry.bioturing.com
helm upgrade --install --set secret.data.bbtoken="${BBTOKEN}" \
 --set secret.data.domain="${SVHOST}" \
 --set secret.server.certificate="${SSLCRT}" \
 --set secret.server.key="${SSLKEY}" \
 --set secret.server.useletsencrypt="${USELETSENCRYPT}" \
 --set secret.server.lcall="${LC_ALL}" \
 --set secret.server.lclang="${LC_LANG}" \
 --set secret.admin.username="${ADMIN_USERNAME}" \
 --set secret.admin.password="${ADMIN_PASSWORD}" \
bioturing bioturing/ecosystem --version ${CHART_VERSION}
```

4. Notices

4.1. Security

- BioTuring System uses HTTPS protocol to securely communicate over the network.
- All of the users need to authenticate using a BioTuring account or the company's SSO to access the platform.
- We highly recommend setting up a private VPC network for IP restriction.
- The data stays behind the company firewall.
- BioTuring System does not track any usage logs.

4.2. Data visibility

- Data can be uploaded to Personal Workspace or Data Sharing group.
- In the Personal Workspace, only the owner can see and manipulate the data she/he uploaded.
- In the Data Sharing group, only people in the group can see the data.
- In the Data Sharing group, only people with sufficient permissions can manipulate the data.