

# Running the Sandbox API with Minikube

This guide explains:

-  Will this API work with Minikube?
  -  How to set up Minikube
  -  How to build images correctly
  -  Required permissions (RBAC)
  -  How to test everything locally
- 

## Will This API Work with Minikube?

Yes — **100% it will work.**

Minikube runs a real Kubernetes cluster locally. Your API talks to the Kubernetes API server using:

```
@kubernetes/client-node
```

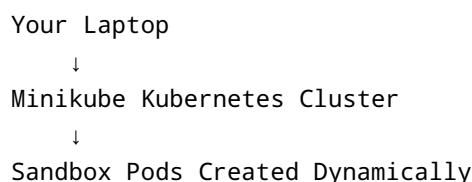
As long as:

- Minikube is running
- Your kubeconfig is configured
- The sandbox image exists inside Minikube

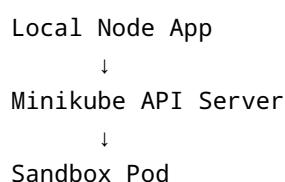
Your API can dynamically create Pods.

---

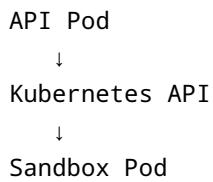
## Architecture When Using Minikube



If API runs locally:



If API runs inside Minikube:



Both approaches work.

---

## Step-by-Step Setup

---

### Install Minikube

Install from official website:

<https://minikube.sigs.k8s.io/docs/start/>

---

### Start Minikube

```
minikube start
```

Verify:

```
kubectl get nodes
```

You should see one node running.

---

### Make Docker Use Minikube's Docker Engine

Very important step.

```
eval $(minikube docker-env)
```

Now when you run `docker build`, it builds inside Minikube.

---

## Build Sandbox Image

```
docker build -t custom-node-sandbox:latest .
```

If you skip this, your pods will fail with:

```
ImagePullBackOff
```

---

## Run Your API

If running locally:

```
npm run dev
```

Your code will use:

```
kc.loadFromDefault();
```

Which loads:

```
~/.kube/config
```

That config automatically points to Minikube.

---

## Testing Pod Creation

Call your endpoint:

```
POST /api/v1/sandbox/init
```

Then check:

```
kubectl get pods
```

You should see:

sandbox-1-abc123

## If Running API Inside Minikube

You must create RBAC role.

Example Role:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: sandbox-role
rules:
- apiGroups: [""]
  resources: ["pods", "pods/exec"]
  verbs: ["create", "delete", "get", "list"]
```

And bind it to your ServiceAccount.



## Important: Wait for Pod Readiness

Pods take time to start.

Before executing code, you should:

- Check Pod phase
- Wait until status = Running

Otherwise exec may fail.



## Common Errors and Fixes



### ImagePullBackOff

Build image inside Minikube using docker-env.



### Forbidden Error

Missing RBAC permissions.

## Exec Fails

Pod not ready yet.

## Pod Pending

Not enough memory allocated to Minikube.

Fix by increasing memory:

```
minikube start --memory=4096
```

---

## Resource Requirements

Each sandbox pod uses:

- 512Mi memory
- 500m CPU

If MAX\_CONTAINERS = 3

Minimum recommended Minikube memory:

2GB - 4GB

---

## Final Answer

Yes, your Kubernetes-based API works perfectly with Minikube.

Minikube is ideal for:

- Local development
- Testing dynamic pod creation
- Debugging sandbox execution
- Learning Kubernetes

It behaves like a real cluster, just single-node.

---

# Production vs Minikube

Feature	Minikube	Production Cluster
Nodes	1	Many
Scaling	Manual	Auto-scaling
Load balancing	Basic	Advanced
Suitable for	Development	Production

---

## Final Architecture Summary

Your API does NOT create Docker containers directly anymore.

It:

- Talks to Kubernetes API
- Kubernetes schedules Pods
- Pods run your sandbox image
- You exec into Pods
- You delete Pods after use

That works locally in Minikube and in real cloud clusters.

---

You are now building cloud-native infrastructure 