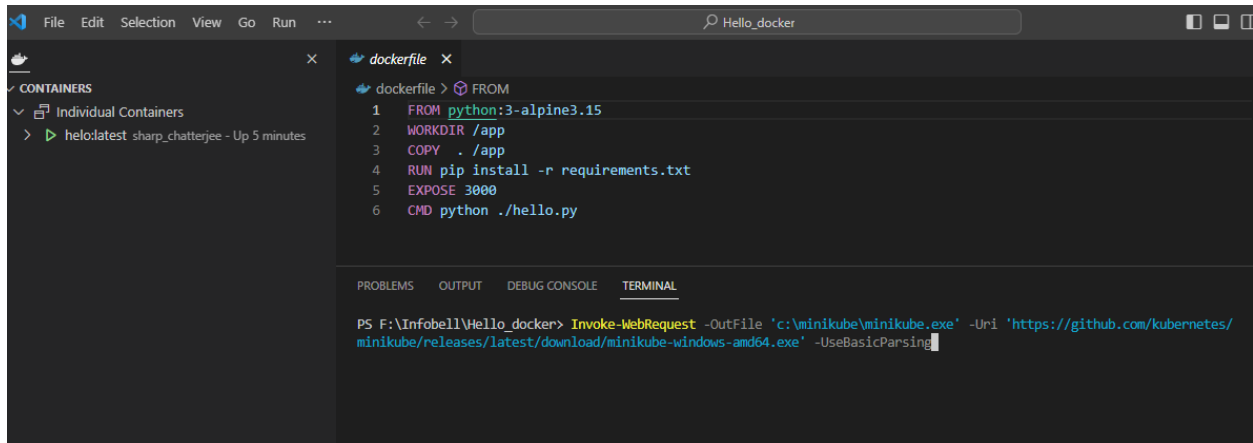


Minikube Installation and Commands

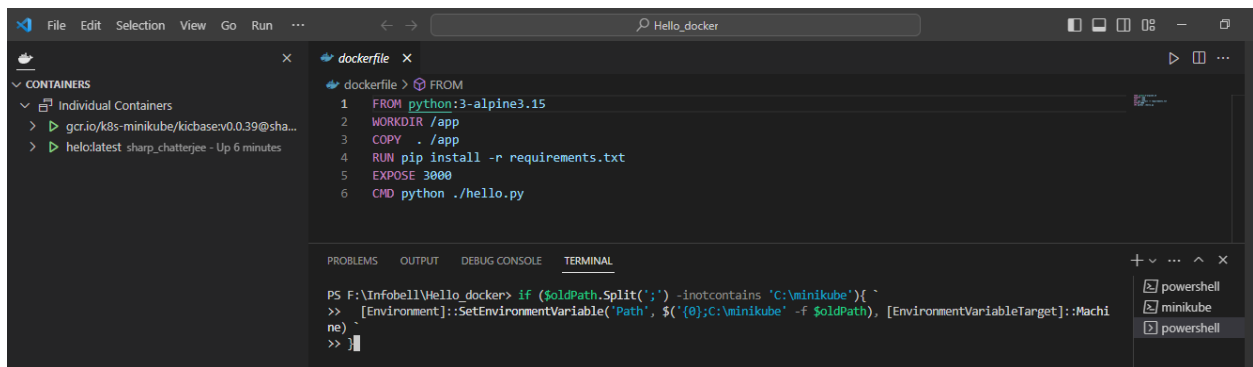


The screenshot shows the Visual Studio Code interface. On the left, the 'CONTAINERS' sidebar is open, showing a list of containers. The main editor area displays a Dockerfile with the following content:

```
dockerfile > FROM
1 FROM python:3-alpine3.15
2 WORKDIR /app
3 COPY . /app
4 RUN pip install -r requirements.txt
5 EXPOSE 3000
6 CMD python ./hello.py
```

Below the Dockerfile, the 'TERMINAL' tab is active, showing the following command being executed:

```
PS F:\Infobell\Hello_docker> Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe' -UseBasicParsing
```



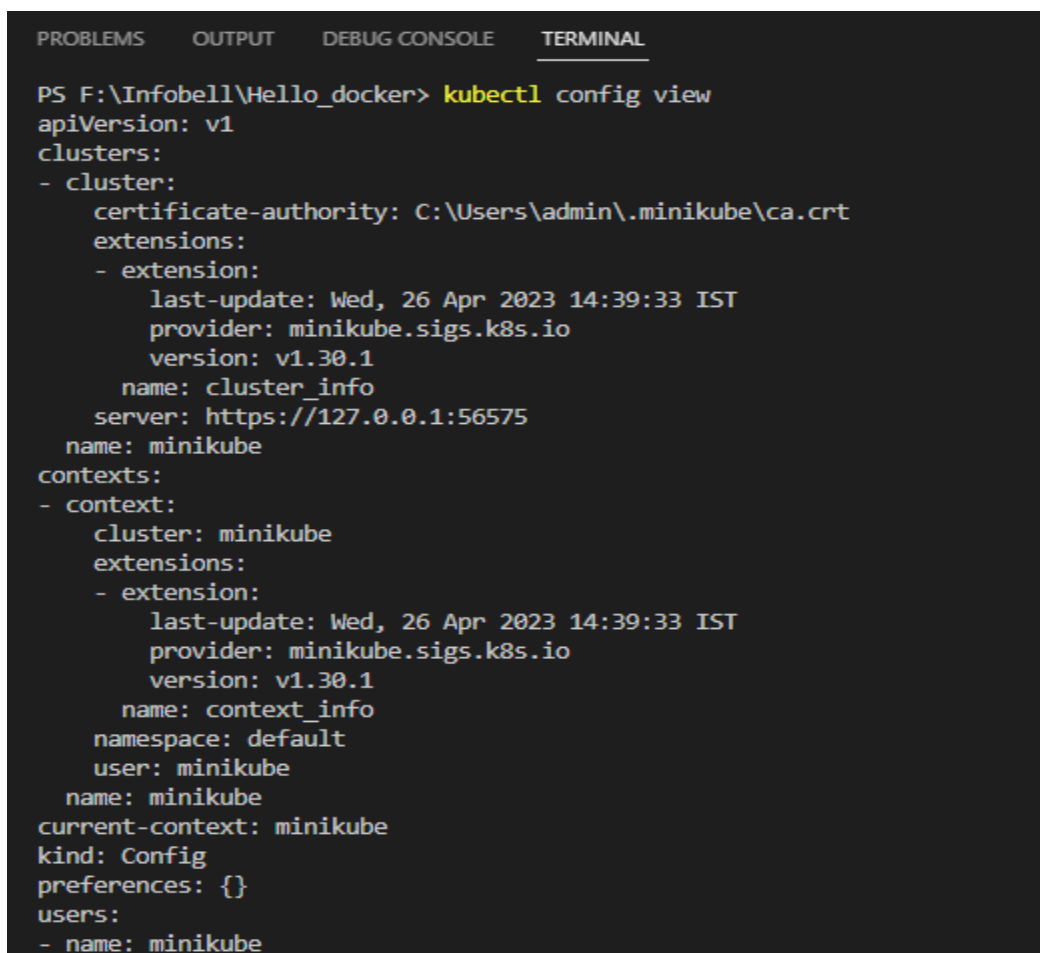
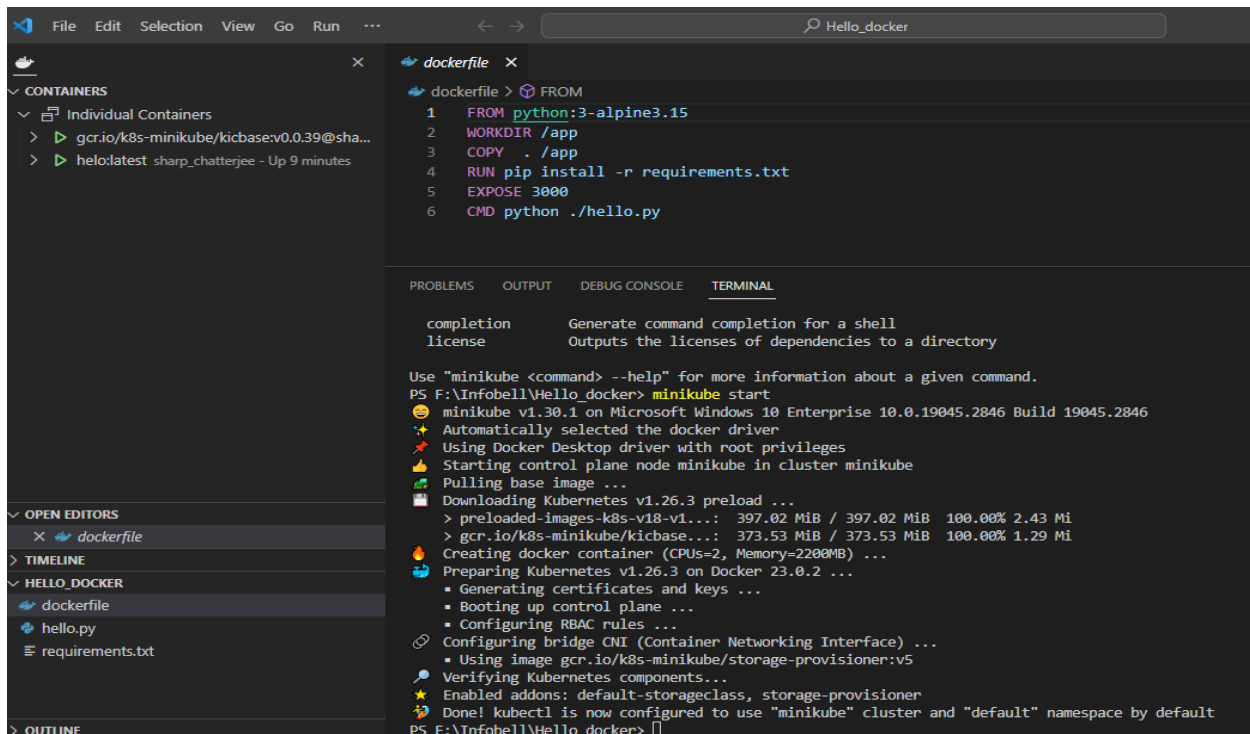
The screenshot shows the Visual Studio Code interface. On the left, the 'CONTAINERS' sidebar is open, showing a list of containers. The main editor area displays a Dockerfile with the following content:

```
dockerfile > FROM
1 FROM python:3-alpine3.15
2 WORKDIR /app
3 COPY . /app
4 RUN pip install -r requirements.txt
5 EXPOSE 3000
6 CMD python ./hello.py
```

Below the Dockerfile, the 'TERMINAL' tab is active, showing the following command being executed:

```
PS F:\Infobell\Hello_docker> if ($oldPath.Split(';') -notcontains 'C:\minikube'){ `
>> [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f $oldPath), [EnvironmentVariableTarget]::Machi
>> ne) `
>> }
```

```
PS F:\Infobell\Hello_docker> minikube version
minikube version: v1.30.1
commit: 08896fd1dc362c097c925146c4a0d0dac715ace0
```



See 'kubectl get --help' for usage.

```
PS F:\Infobell\Hello_docker> kubectl get --help
```

Display one or many resources.

Prints a table of the most important information about the specified resources. You can filter the list using a label selector and the --selector flag. If the desired resource type is namespaced you will only see results in your current namespace unless you pass --all-namespaces.

By specifying the output as 'template' and providing a Go template as the value of the --template flag, you can filter the attributes of the fetched resources.

Use "kubectl api-resources" for a complete list of supported resources.

Examples:

List all pods in ps output format

```
kubectl get pods
```

List all pods in ps output format with more information (such as node name)

```
kubectl get pods -o wide
```

List a single replication controller with specified NAME in ps output format

```
kubectl get replicationcontroller web
```

List deployments in JSON output format, in the "v1" version of the "apps" API group

```
kubectl get deployments.v1.apps -o json
```

List a single pod in JSON output format

```
kubectl get -o json pod web-pod-13je7
```

List a pod identified by type and name specified in "pod.yaml" in JSON output format

Use "kubectl options" for a list of global command-line options (applies to all commands).

```
PS F:\Infobell\Hello_docker> kubectl get deploy --A
```

error: unknown flag: --A

See 'kubectl get --help' for usage.

```
PS F:\Infobell\Hello_docker> kubectl get deploy -A
```

NAMESPACE	NAME	READY	UP-TO-DATE	AVAILABLE	AGE
default	helo	0/1	1	0	2m24s
kube-system	coredns	1/1	1	1	5m38s

```
PS F:\Infobell\Hello_docker> █
```

```
PS F:\Infobell\Hello_docker> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
helo-7fc7874997-5t864	0/1	ImagePullBackOff	0	10m

```
PS F:\Infobell\Hello_docker> █
```