



Web



Deployment



WEB &gt; GUIDES &gt; PSPDFKIT SERVER &gt; DEPLOYMENT

# Deploying PSPDFKit Server to Microsoft Azure



PSPDFKit Server has been deprecated and replaced by [Document Engine](#). To migrate to Document Engine and unlock advanced document processing capabilities, refer to our [migration guide](#). Learn more about these enhancements on our [blog](#).

This guide will walk you through the steps for deploying PSPDFKit Server to the Microsoft Azure Container Service with [Kubernetes](#).

## Setting up Azure CLI

To be able to deploy PSPDFKit Server to the [Microsoft Azure Container Service](#) with [Kubernetes](#), you have to set up the [Azure CLI](#) utility to manage your [Kubernetes](#) cluster in the command line.

To install [Azure CLI](#), follow the installation instructions from the [Azure CLI installation guide](#).

After you've installed [Azure CLI](#), run the following command to log in to [Microsoft Azure](#):

```
az login
```



This command will print the URL <https://microsoft.com/devicelogin> and the code for signing in. Open the URL in your browser and enter the code to sign in to your [Microsoft Azure](#) account.



ASK AI



## Device Login

Enter the code that you received from the application on your device

©2018 Microsoft  
[Terms of use](#) [Privacy & cookies](#)



## Creating a resource group

To create a resource group, run the following:

```
az group create -l eastus -n pspdfkitresourcegroup
```



In this example, we created the resource group in the `eastus` region with the name `pspdfkitresourcegroup`. An overview of available regions can be found on Microsoft's [Azure Regions](#) page.

## Creating a Kubernetes cluster

To be able to manage your [Kubernetes](#) cluster from the command line, you have to install `kubectl`:

```
az aks install-cli
```



To create a [Kubernetes](#) cluster with the name `pspdfkitAKScluster`, run the following:

To connect `kubectl` with your cluster, execute:

```
az aks get-credentials -g pspdfkitresourcegroup -n pspdfkitAKScluster
```



## Creating a ConfigMap

ConfigMaps allow you to decouple configuration artifacts from image content. To create the `pspdfkit-config` ConfigMap, run the following command:

```
kubectl create configmap pspdfkit-config
```



After the ConfigMap is created, you can edit it with the following:

```
kubectl edit configmap pspdfkit-config
```



This should open the created ConfigMap in your editor. Edit the file to match the following file and replace `activation_key` with your activation key:

```
1  # Please edit the object below. Lines beginning with a '#' will be ignored,
2  # and an empty file will abort the edit. If an error occurs while saving,
3  # this file will be reopened with the relevant failures.
4  #
5  apiVersion: v1
6  data:
7    activation_key: YOUR_ACTIVATION_KEY_GOES_HERE
8    api_auth_token: secret
9    dashboard_password: secret
10   dashboard_username: dashboard
11   jwt_algorithm: RS256
12   jwt_public_key: |
13     -----BEGIN PUBLIC KEY-----
14     MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2gzhmJ9TDanEzWdP1WG+
15     0Ecwbe7f3bv6e5UUpvcT5q68IQJKP47AQdBANsLFVi4X9SaurbWoXdS6jpmPpk24
16     QvitzLNFphHdwjFBeLTaOa6taZrSusoFvrtK9x5sW4zzt/bkpUraNx82Z8MwLwr
17     t6HlY7dgO9+xBAabj4t1d2t+0HS8O/ed3CB6T2lj6S8AbLDSEFc9ScO6Uc1XJlSo
18     rgyJJSPCpNhSq3AubEZ1wMSliEtgAzTPRDsQv50qWIbn634HLWxTP/UH6YNJBwzt
19     3O6q29kTtjXlMGXCvin37PyX4JylIiPFwJm45aWJGKSfVGMDojTJbuUtM+8P9Rrn
```



```

20     AwIDAQAB
21     -----END PUBLIC KEY-----
22     pgdata: /var/lib/postgresql/data/pgdata
23     pgdatabase: pspdfkit
24     pghost: pspdfkitdb
25     pgpassword: examplepassword
26     pgport: "5432"
27     pguser: pspdfkit
28     secret_key_base: secret-key-base
29     kind: ConfigMap

```

Don't change anything that comes after the `kind: ConfigMap` line, because that part is autogenerated.

## Creating an Azure Disk Volume

To create the [Azure Disk Volume](#) where the database saves the data, run:

```
az group list --output table
```



This will print a list of resource groups. Copy the name of the resource group that starts with `MC` — for example, `MC_pspdfkitresourcegroup_pspdfkitAKScluster_eastus` — and insert it into the following command after `--resource-group`:

```

1  az disk create \
2    --resource-group MC_pspdfkitresourcegroup_pspdfkitAKScluster_eastus \
3    --name postgres-disk\
4    --size-gb 50 \
5    --query id --output tsv

```



This will print the disk ID of the created disk. For example:

```
/subscriptions/12345abc-def1-2345-6789-abcdef123456/resourceGroups/MC_pspdfk:
```



## Creating Services and Deployments

[Kubernetes Services](#) and [Deployments](#) can be configured in a file. To run PSPDFKit Server, you have to define a Service and a Deployment for the database and a Service and a Deployment for PSPDFKit Server. First, create the file that configures the database Service and Deployment. To do this, create the

`pspdfkitdb.yml` file in the current directory:

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: pspdfkitdb
5  spec:
6    ports:
7      - port: 5432
8    selector:
9      app: pspdfkitdb
10 ---
11 apiVersion: apps/v1
12 kind: Deployment
13 metadata:
14   name: pspdfkitdb
15 spec:
16   selector:
17     matchLabels:
18       app: pspdfkitdb
19   template:
20     metadata:
21       labels:
22         app: pspdfkitdb
23     spec:
24       containers:
25         - image: "postgres:15"
26           name: pspdfkitdb
27           env:
28             - name: POSTGRES_USER
29               valueFrom:
30                 configMapKeyRef:
31                   name: pspdfkit-config
32                   key: pguser
33             - name: POSTGRES_PASSWORD
34               valueFrom:
35                 configMapKeyRef:
36                   name: pspdfkit-config
37                   key: pgpassword
38             - name: PGDATA
39               valueFrom:
40                 configMapKeyRef:
41                   name: pspdfkit-config
42                   key: pgdata
43       ports:
44         - containerPort: 5432
45         name: pspdfkitdb
46       volumeMounts:
47         - name: postgres-storage
```

```

48         mountPath: /var/lib/postgresql/data
49     volumes:
50     - name: postgres-storage
51       azureDisk:
52         kind: Managed
53         diskName: postgres-disk
54         diskURI: /subscriptions/12345abc-def1-2345-6789-abcdef123456/resou:

```

Make sure to replace `diskURI` in the last line with the disk ID you got from the command before.

After you've created the configuration file for the database, create the configuration for PSPDFKit Server ( `pspdfkit.yml` ) and ensure that the `pspdfkit/pspdfkit` image tag corresponds to the latest PSPDFKit Server version:

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: pspdfkit
5  spec:
6    ports:
7      - protocol: TCP
8        port: 5000
9        targetPort: 5000
10   selector:
11     app: pspdfkit
12   type: LoadBalancer
13 ---
14 apiVersion: apps/v1
15 kind: Deployment
16 metadata:
17   name: pspdfkit
18 spec:
19   selector:
20     matchLabels:
21       app: pspdfkit
22   template:
23     metadata:
24       labels:
25         app: pspdfkit
26     spec:
27       containers:
28       - image: "pspdfkit/pspdfkit:1.4.1"
29         name: pspdfkit
30         env:
31         - name: PGUSER
32           valueFrom:
33             configMapKeyRef:
34               name: pspdfkit-config
35               key: pguser
36         - name: PGPASSWORD
37           valueFrom:

```



```

38         configMapKeyRef:
39             name: pspdfkit-config
40             key: pgpassword
41     - name: PGDATABASE
42       valueFrom:
43         configMapKeyRef:
44             name: pspdfkit-config
45             key: pgdatabase
46     - name: PGHOST
47       valueFrom:
48         configMapKeyRef:
49             name: pspdfkit-config
50             key: pgghost
51     - name: PGPORT
52       valueFrom:
53         configMapKeyRef:
54             name: pspdfkit-config
55             key: pgport
56     - name: ACTIVATION_KEY
57       valueFrom:
58         configMapKeyRef:
59             name: pspdfkit-config
60             key: activation_key
61     - name: API_AUTH_TOKEN
62       valueFrom:
63         configMapKeyRef:
64             name: pspdfkit-config
65             key: api_auth_token
66     - name: SECRET_KEY_BASE
67       valueFrom:
68         configMapKeyRef:
69             name: pspdfkit-config
70             key: secret_key_base
71     - name: JWT_ALGORITHM
72       valueFrom:
73         configMapKeyRef:
74             name: pspdfkit-config
75             key: jwt_algorithm
76     - name: JWT_PUBLIC_KEY
77       valueFrom:
78         configMapKeyRef:
79             name: pspdfkit-config
80             key: jwt_public_key
81     - name: DASHBOARD_USERNAME
82       valueFrom:
83         configMapKeyRef:
84             name: pspdfkit-config
85             key: dashboard_username
86     - name: DASHBOARD_PASSWORD
87       valueFrom:
88         configMapKeyRef:
89             name: pspdfkit-config
90             key: dashboard_password
91 ports:

```



```
92         - containerPort: 5000
93         name: pspdfkit
```

To create all Services and Deployments needed to run PSPDFKit Server, execute:

```
kubectl create -f ./pspdfkitdb.yml
```

```
kubectl create -f ./pspdfkit.yml
```

## Viewing the dashboard

To be able to access the server, you have to get the external IP address that was assigned to the server. Run the following command to view all the Services in your cluster, along with their assigned external IP addresses:

```
kubectl get services
```

This will show something like the following:

	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
1	kubernetes	ClusterIP	10.15.240.1	<none>	443/TCP
2	pspdfkit	LoadBalancer	10.15.247.197	12.345.678.910	5000:32393/TCP
3	pspdfkitdb	ClusterIP	10.15.244.127	<none>	5432/TCP

Copy the `EXTERNAL-IP` address from the `pspdfkit` column and access the [dashboard](#) with the port `5000` and the `/dashboard` path in your web browser. In this example, you would access the dashboard with `http://12.345.678.910:5000/dashboard`.

## Limitations

Be aware that this is just an example setup, and we recommend looking deeper into the Microsoft Azure Container Service for a production-ready setup.



---

Was this helpful?

 YES

 NO

---

Questions? [Contact us](#)

