

AZURE

The screenshot shows the Microsoft Azure Cost Management Overview page. On the left, there's a sidebar with navigation links for Cost Management, Billing, Products + services, and Settings. The main area has a heading 'Analyze and optimize cloud costs' with three cards: 'Setup your account', 'Report on and analyze trends', and 'Control and optimize costs'. A top right corner shows a message about a \$200.00 credit remaining.

First of all you need to switch subscription in default directory for apply free billing 200 dollars !!!!!!!!

Prerequisites

Create azure subscription

The screenshot shows the Microsoft Azure Subscriptions blade. It displays a table with one row for 'Azure subscription 1'. The columns include Subscription name, Subscription ID, My role, Current cost, Secure Score, Parent management group, and Status. The status is marked as 'Active'.

Subscription name	Subscription ID	My role	Current cost	Secure Score	Parent management group	Status
Azure subscription 1	b124b468-39ee-416c-a1ef-9738f476fa3c	Owner	Not available	-		Active

Create azure devops organization

The screenshot shows the Microsoft Azure Subscriptions blade, identical to the previous one, displaying a table with one row for 'Azure subscription 1'. The status is marked as 'Active'.

Subscription name	Subscription ID	My role	Current cost	Secure Score	Parent management group	Status
Azure subscription 1	b124b468-39ee-416c-a1ef-9738f476fa3c	Owner	Not available	-		Active

Installing Terraform and Azure Cli

1 Download to any directory

2 Add path in system env this folder

C:\Users\Nik>terraform version

Terraform v1.3.4

on windows_amd64

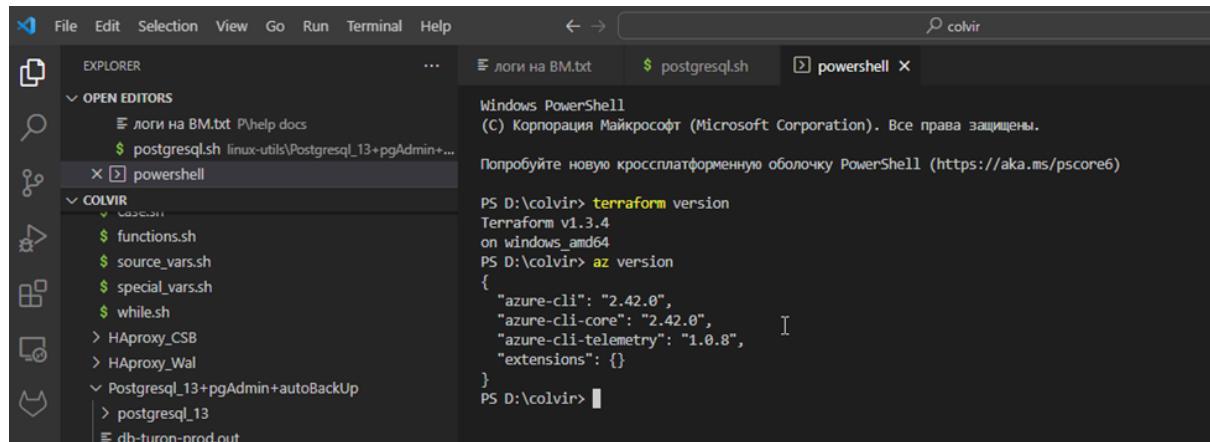
3 Download and install Azure CLI

C:\Users\Nik>az version

{

```
"azure-cli": "2.42.0",
"azure-cli-core": "2.42.0",
"azure-cli-telemetry": "1.0.8",
"extensions": {}
```

}



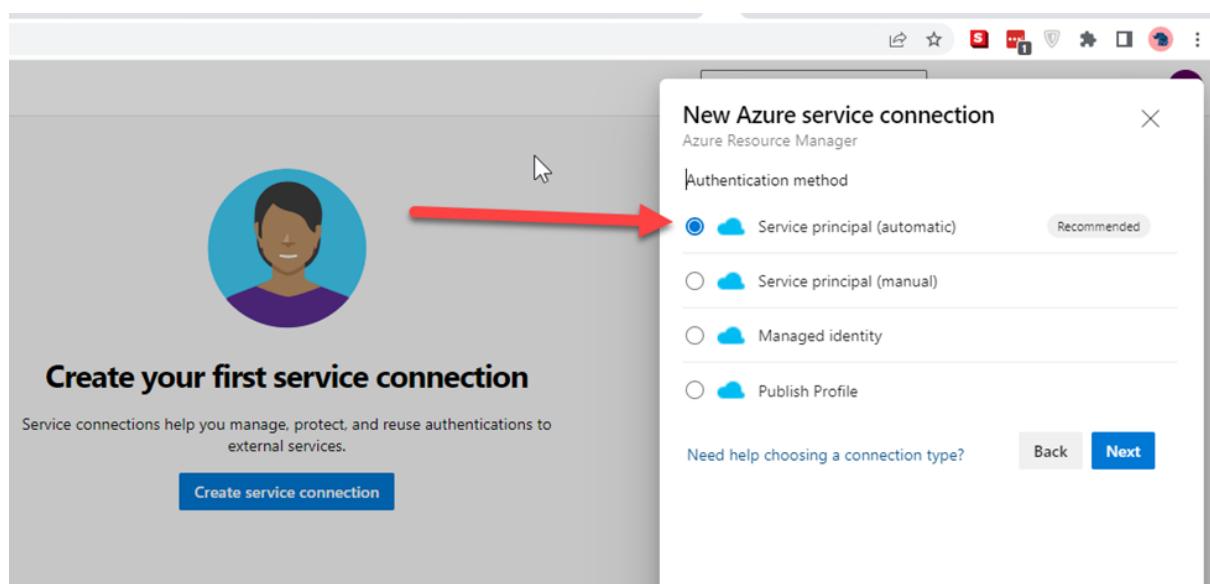
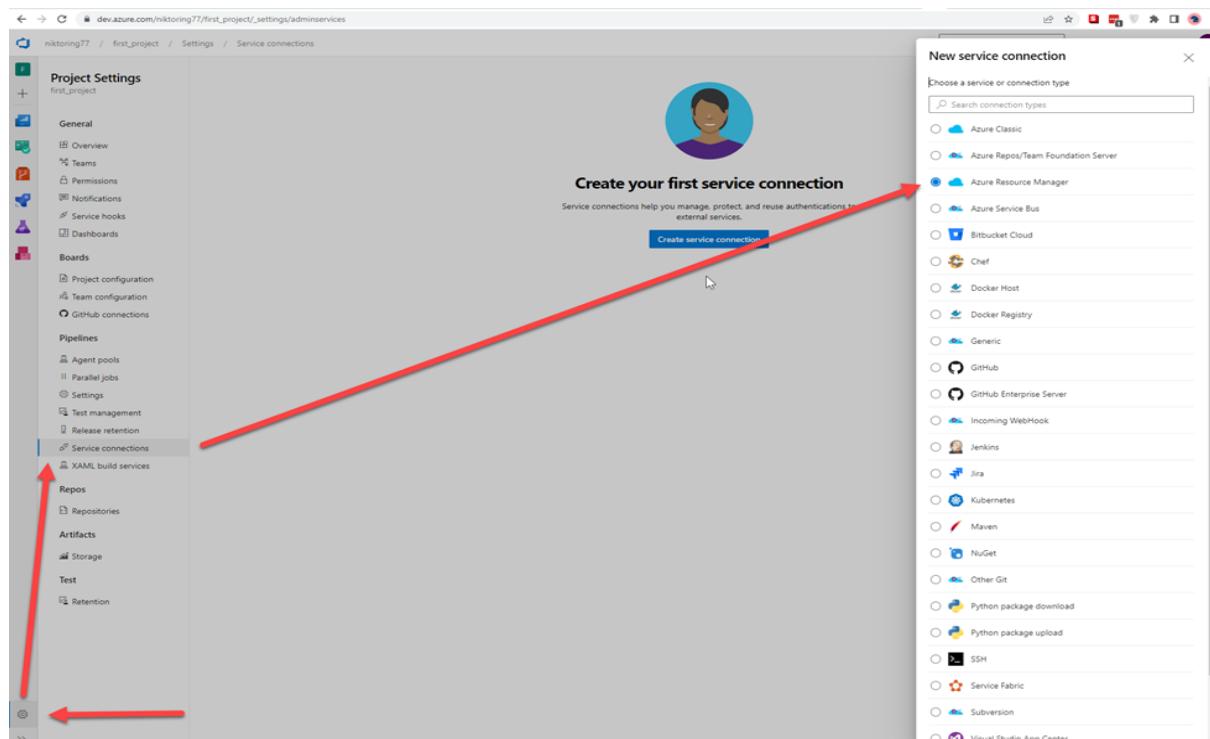
The screenshot shows a Windows terminal window with three tabs: 'логи на ВМ.txt', '\$ postgresql.sh', and '\$ powershell'. The '\$ powershell' tab is active and displays the following command history:

```
PS D:\colvir> terraform version
Terraform v1.3.4
on windows_amd64
PS D:\colvir> az version
{
  "azure-cli": "2.42.0",
  "azure-cli-core": "2.42.0",
  "azure-cli-telemetry": "1.0.8",
  "extensions": {}}
```

Homework

Part 1 – Configure application

1. Create a service connection in a Azure DevOps project to your subscription -
<https://learn.microsoft.com/en-us/azure/devops/pipelines/library/service-endpoints?view=azure-devops&tabs=yaml>





Create your first service connection

Service connections help you manage, protect, and reuse authentications to external services.

[Create service connection](#)

New Azure service connection

Azure Resource Manager using service principal (automatic)

Scope level

Subscription

Management Group

Machine Learning Workspace

Subscription

Azure subscription 1 (b124b468-39ee-416c-a1ef-9738f476fa3c) ▾

Resource group

Details

Service connection name

My_Subscription

Description (optional)

My_Subscription for first .net project

Security

Grant access permission to all pipelines

[Learn more](#)

[Troubleshoot](#)

[Back](#) [Save](#)

← → C dev.azure.com/niktorin77/first_project/_settings/adminservices?resourceId=1340354b-a6c6-4a92-aa29-03e8b3dd863a

niktorin77 / first_project / Settings / Service connections

Project Settings first_project

General

- Overview
- Teams
- Permissions
- Notifications
- Service hooks
- Dashboards

Boards

- Project configuration
- Team configuration
- GitHub connections

Pipelines

- Agent pools
- Parallel jobs
- Settings
- Test management
- Release retention

Service connections

XAML build services

Repos

Repositories

Artifacts

Storage

Test

Retention

← My_Subscription

Overview Usage history

Details

Service connection type

Azure Resource Manager using service principal authentication

Manage service connection roles

Manage Service Principal

Creator

My_Subscription for first .net project

4 Create an Azure DevOps repo -
<https://learn.microsoft.com/en-us/azure/devops/repos/git/create-new-repo?view=azure-devops> You can use import repository to import from existing source control version like github

Add to Azure repo my project

Name	Last change	Commits
devops	11h ago	81705e8b part of devOps file nlistopad
src	Yesterday	08452853 Init nlistopad
.browserslistrc	Yesterday	08452853 Init nlistopad
.editorconfig	Yesterday	08452853 Init nlistopad
.gitignore	Yesterday	08452853 Init nlistopad
angular.json	11h ago	81705e8b part of devOps file nlistopad
JS karma.conf.js	Yesterday	08452853 Init nlistopad
package-lock.json	11h ago	81705e8b part of devOps file nlistopad
package.json	11h ago	81705e8b part of devOps file nlistopad
README.md	Yesterday	08452853 Init nlistopad
tsconfig.app.json	Yesterday	08452853 Init nlistopad
tsconfig.json	Yesterday	08452853 Init nlistopad
tsconfig.spec.json	Yesterday	08452853 Init nlistopad

2) Find a .net pet project for the experiments. I find my Angular project

Create pipeline

Create your first Pipeline
Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.

Create Pipeline

Azure DevOps niktoring77 / first_project / Pipelines

first_project

- Overview
- Boards
- Repos
- Pipelines**
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups
- Test Plans
- Artifacts

Connect **Select** **Configure**

New pipeline

Where is your code?

- Azure Repos Git YAML
Free private Git repositories, pull requests, and code search
- Bitbucket Cloud YAML
Hosted by Atlassian
- Github YAML
Home to the world's largest community of developers
- Github Enterprise Server YAML
The self-hosted version of GitHub Enterprise
- Other Git
Any generic Git repository
- Subversion
Centralized version control by Apache

Use the classic editor to create a pipeline without YAML.

Azure DevOps niktoring77 / first_project / Pipelines

first_project

- Overview
- Boards
- Repos
- Pipelines**
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups
- Test Plans
- Artifacts

Select your repository

Tell us where your sources are.
You can customize how to get these sources from the repository later.

Select a source

- Azure Repos Git
- Github
- Github Enterprise Server

Team project: first_project

Repository: first_project

Default branch for manual and scheduled builds: master

Continue

Azure DevOps niktoring77 / first_project / Pipelines

first_project

- Overview
- Boards
- Repos
- Pipelines**
- Pipelines
- Environments

Select a template

Or start with an **Empty job**

Configuration as code

YAML

Looking for a better experience to configure your pipelines using YAML files? Try the new YAML pipeline creation experience. [Learn more](#)

Featured

.NET Desktop

Build and test a .NET or Windows classic desktop solution.

Azure DevOps niktoring77 / first_project / Pipelines

first_project

- Overview
- Boards
- Repos
- Pipelines**
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups

... > first_project-Cl

Tasks Variables Triggers Options History Save & queue Discard Summary Queue ...

Pipeline Build pipeline

Get sources first_project master

Agent job 1 Run on agent

Add tasks Refresh

web app

Deploy Azure Static Web App (PREVIEW) Build and deploy an Azure Static Web App

Azure Web App for Containers Deploy containers to Azure App Service

Azure Web App Deploy an Azure Web App for Linux or Windows

Azure App Service Settings Update/Add App settings an Azure Web App for Linux or Windows

+ Add

Azure DevOps Pipeline configuration for 'first_project-Cl'. The 'Azure Web App Deploy' task is selected. The 'View YAML' button is highlighted with a red box and an arrow.

The 'Copy to clipboard' modal window displays the following YAML code:

```

steps:
- task: AzureWebApp@1
  displayName: 'Azure Web App Deploy: webApp-E pam'
  inputs:
    azureSubscription: 'My_Subscription'
    appType: 'webApp'
    appName: 'webApp-E pam'
  
```

The 'Copy to clipboard' modal window displays the following YAML code:

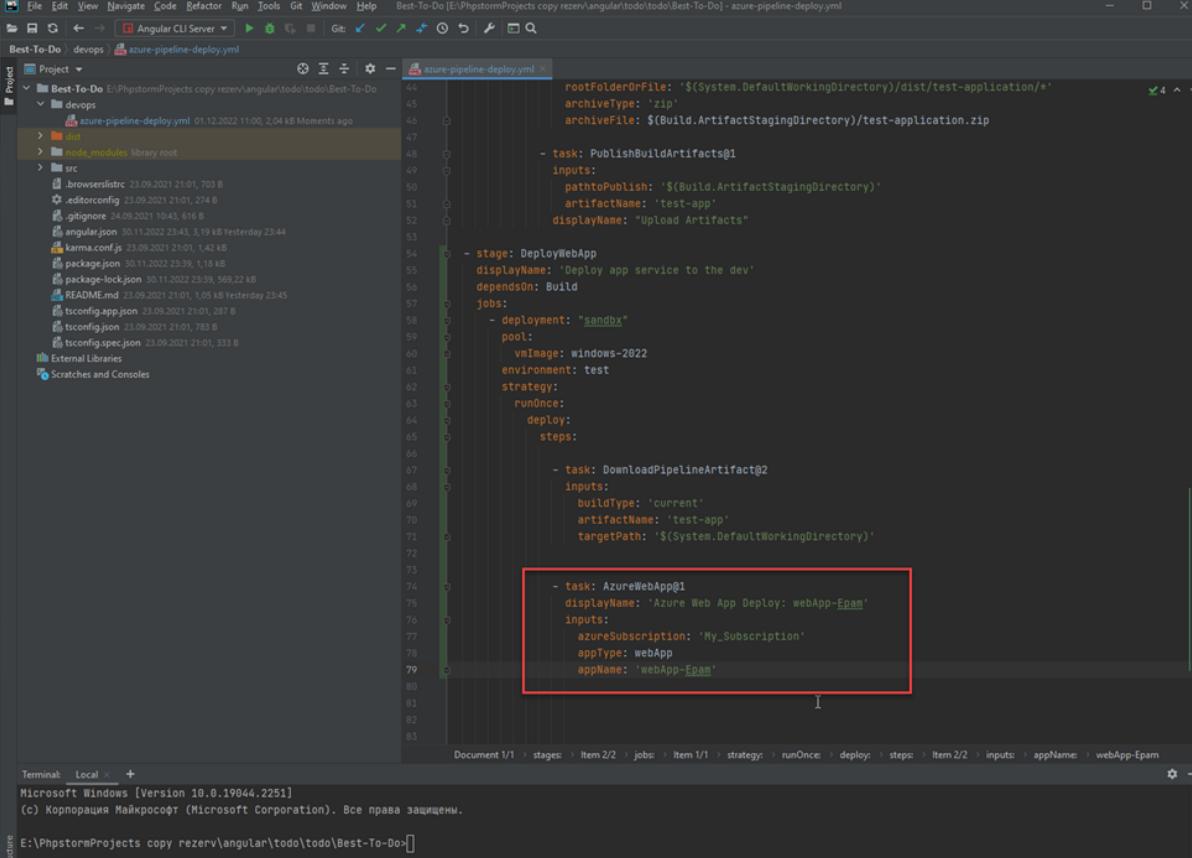
```

steps:
- task: AzureRmWebAppDeployment@4
  displayName: 'Azure App Service Deploy: webApp-E pam'
  inputs:
    azureSubscription: 'Azure_subscription_1 (b124b468-39ee-416c-a1ef-9738f476fa3c)'
    WebAppName: 'webApp-E pam'
  
```

In the background, the 'Azure App Service Deploy' task configuration is shown. The 'Azure subscription' dropdown is highlighted with a red box. A red arrow points from the highlighted value in the modal to this dropdown. Another red arrow points from the highlighted value to the 'View YAML' button.

CREATE or APPROVE your subscribe !!!!!!

Add part of code to project



```
rootFolderOrFile: '$(System.DefaultWorkingDirectory)/dist/test-application/'
archiveType: 'zip'
archiveFile: '$(Build.ArtifactStagingDirectory)/test-application.zip'

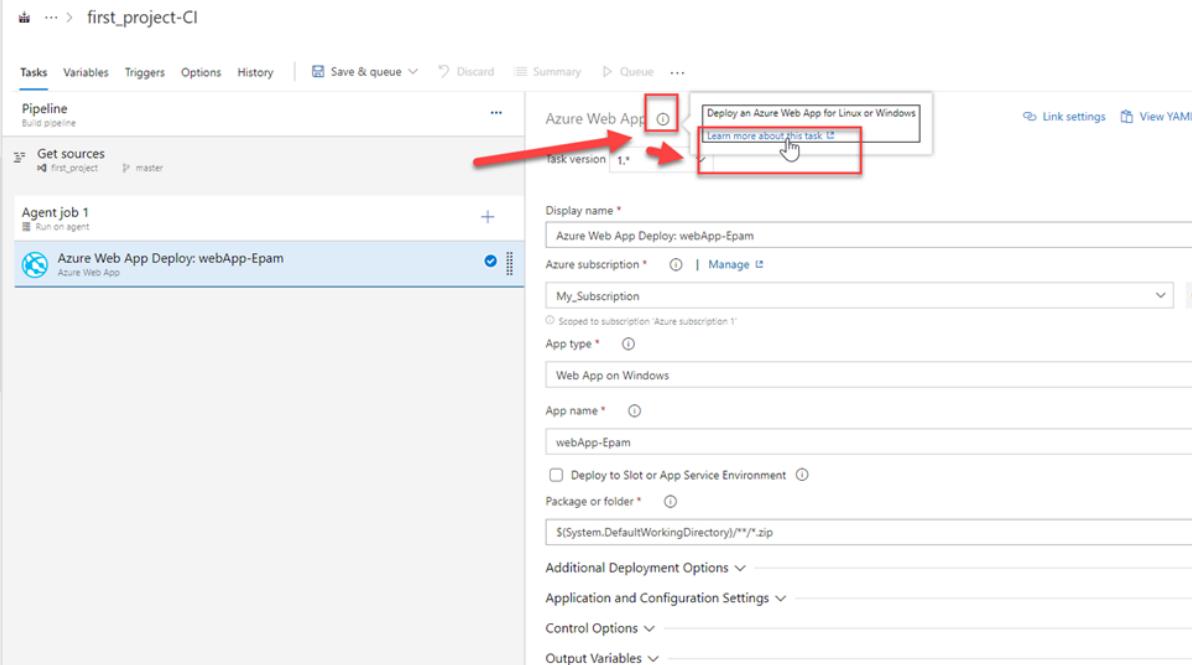
- task: PublishBuildArtifacts@1
  inputs:
    pathToPublish: '$(Build.ArtifactStagingDirectory)'
    artifactName: 'test-app'
  displayName: 'Upload Artifacts'

  stage: DeployWebApp
  displayName: 'Deploy app service to the dev'
  dependsOn: Build
  jobs:
    - deployment: "sandbox"
      pool:
        vmImage: windows-2022
        environment: test
      strategy:
        runOnce:
          deploy:
            steps:
              - task: DownloadPipelineArtifact@2
                inputs:
                  buildType: 'current'
                  artifactName: 'test-app'
                  targetPath: '$(System.DefaultWorkingDirectory)'

              - task: AzureWebApp@1
                displayName: 'Azure Web App Deploy: webApp-Epam'
                inputs:
                  azureSubscription: 'My_Subscription'
                  appType: webApp
                  appName: 'webApp-Epam'
```

+ We need to add some parameters

Read in manual



The screenshot shows the Azure DevOps pipeline editor. On the left, there's a pipeline named 'first_project-Cl' with a 'Get sources' step and an 'Agent job 1' step containing an 'Azure Web App Deploy: webApp-Epam' task. The task configuration pane is open on the right. It shows the task name 'Azure Web App' with a help icon, and below it, a dropdown menu showing 'task version 1.*'. A red arrow points to the task name, and another red arrow points to the dropdown menu. The configuration fields include 'Display name' set to 'Azure Web App Deploy: webApp-Epam', 'Azure subscription' set to 'My_Subscription', 'App type' set to 'Web App on Windows', 'App name' set to 'webApp-Epam', and 'Package or folder' set to '\$(System.DefaultWorkingDirectory)/**/*.*zip'.

131 lines (77 sloc) | 12.1 KB

Mountable by the Functions runtime. If this option is in the WorkingFolder become read-only. For more information, see [Run your Azure Functions from a package file.](#)

Parameters of the task

The task is used to deploy a Web project to an existing Azure Web App or Function. The mandatory fields are highlighted with a *.

- Azure Subscription***: Select the AzureRM Subscription. If none exists, then click on the Manage link, to navigate to the Services tab in the Administrators panel. In the tab click on New Service Endpoint and select Azure Resource Manager from the dropdown.
- App Service type***: Select the Azure App Service type. The different app types supported are Function App, Web App on Windows, Web App on Linux, Web App for Containers and Azure App Service Environments
- App Service Name***: Select the name of an existing Azure App Service. Enter the name of the Web App if it was provisioned dynamically using the [Azure PowerShell task](#) and [AzureRM PowerShell scripts](#).
- Deploy to Slot**: Select the option to deploy to an existing slot other than the Production slot. Do not select this option if the Web project is being deployed to the Production slot. The Web App itself is the Production slot.
- Resource Group**: Select the Azure Resource Group that contains the Azure App Service specified above. Enter the name of the Azure Resource Group if has been dynamically provisioned using [Azure Resource Group Deployment task](#) or [Azure PowerShell task](#). This is a required parameter if the option to Deploy to Slot has been selected.
- Slot**: Select the Slot to deploy the Web project to. Enter the name of the Slot if has been dynamically provisioned using [Azure Resource Group Deployment task](#) or [Azure PowerShell task](#). This is a required parameter if the option to Deploy to Slot has been selected.
- Package or Folder***: Location of the Web App zip package or folder on the automation agent or on a UNC path accessible to the automation agent like, \\BudgetIT\Web\Deploy\Fabrikam.zip. Predefined system variables and wild cards like, \$(System.DefaultWorkingDirectory)*.zip can be also used here.
- Select deployment method**: Select the option to choose from auto, zipDeploy and runFromPackage. Default value is Auto-detect where the task tries to select the appropriate deployment technology given the input package, app service type and agent OS.
- Runtime Stack**: Web App on Linux offers two different options to publish your application, one is Custom image deployment (Web App for Containers) and the other is App deployment with a built-in platform image (Web App on Linux). You will see this parameter only when you selected 'Linux Web App' in the App type selection option in the task.



We need to add:

```
package: '$(System.DefaultWorkingDirectory)/**/*.zip'
resourceGroupName: test
```

```
azure-pipeline-deploy.yml
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
```

```
pathToPublish: '$(Build.ArtifactStagingDirectory)'
artifactName: 'test-app'
displayName: "Upload Artifacts"

- stage: DeployWebApp
  displayName: 'Deploy app service to the dev'
  dependsOn: Build
  jobs:
    - deployment: "sandbox"
      pool:
        vmImage: windows-2022
        environment: test
        strategy:
          runOnce:
            deploy:
              steps:

                - task: DownloadPipelineArtifact@2
                  inputs:
                    buildType: 'current'
                    artifactName: 'test-app'
                    targetPath: '$(System.DefaultWorkingDirectory)'

                - task: AzureWebApp@1
                  displayName: 'Azure Web App Deploy: webApp-Epam'
                  inputs:
                    azureSubscription: 'My_Subscription'
                    appType: webApp
                    appName: 'webApp-Epam'
                    package: '$(System.DefaultWorkingDirectory)/**/*.zip'
                    resourceGroupName: test
```

Push changes to Azure repo

The screenshot shows the Azure DevOps interface for a project named "first_project". On the left, the navigation bar includes "Overview", "Boards", "Repos" (selected), "Files", "Commits", "Pushes", "Branches", "Tags", "Pull requests", "Pipelines", "Test Plans", and "Artifacts". The main area displays the repository structure under "first_project/devops". The "azure-pipeline-deploy.yml" file is selected, showing its YAML content:

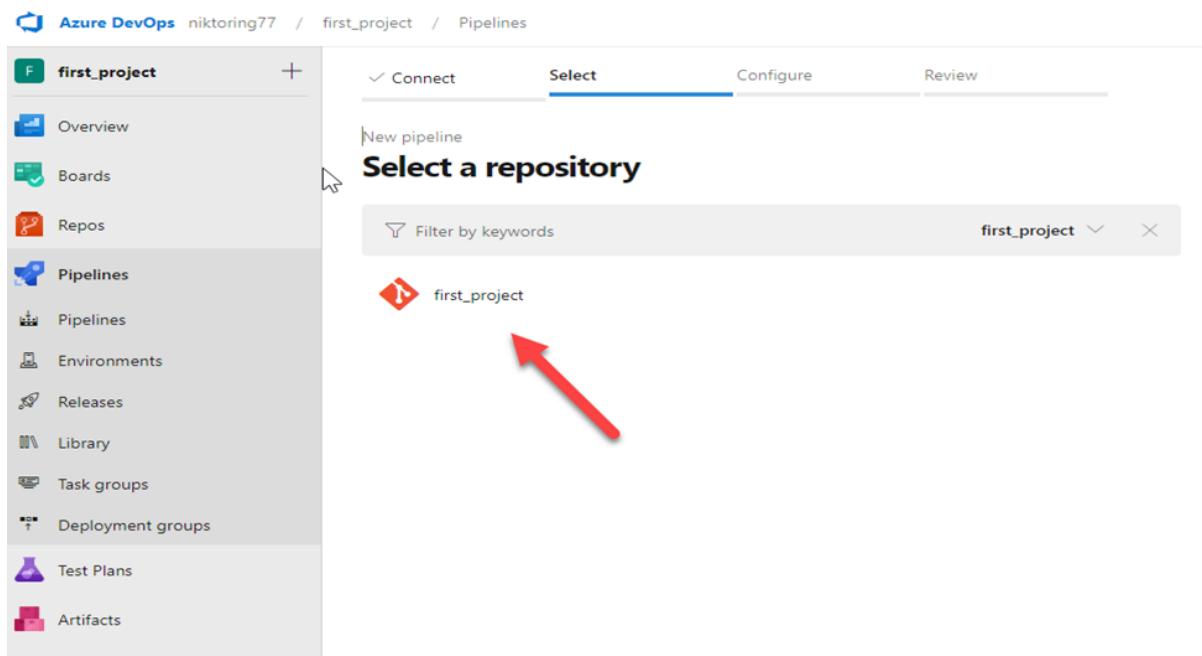
```
environment: validation
strategy:
  runOnce:
    deploy:
      steps:
        - checkout: self
        clean: true
        fetchDepth: 5
      - task: Npm@1
        displayName: 'npm install'
        inputs:
          | verbose: false
      - task: Npm@1
        displayName: 'npm custom'
        inputs:
          | command: custom
          | verbose: false
          | customCommand: 'run build'
      - task: ArchiveFiles@2
        displayName: 'Create an artifact'
        inputs:
          | rootFolderOrFile: '$(System.DefaultWorkingDirectory)/dist/test-application/**'
          | archiveType: 'zip'
          | archiveFile: '$(Build.ArtifactStagingDirectory)/test-application.zip'
      - task: PublishBuildArtifacts@1
        inputs:
          | pathToPublish: '$(Build.ArtifactStagingDirectory)'
          | artifactName: 'test-app'
        displayName: "Upload Artifacts"
    stage: DeployWebapp
    displayName: 'Deploy app service to the dev'
    jobs:
      - deployment: "sandbox"
        pool:
          vmImage: windows-2022
        environment: test
        strategy:
          runOnce:
            deploy:
              steps:
                - task: DownloadPipelineArtifact@2
                  inputs:
                    buildType: 'current'
                    artifactName: 'test-app'
                    targetPath: '$(System.DefaultWorkingDirectory)'
                - task: AzureWebApp@1
                  displayName: 'Azure Web App Deploy: webApp-Epm'
                  inputs:
                    azureSubscription: 'My_Subscription'
                    appType: webApp
                    appName: 'webApp-Epm'
                    package: '$(System.DefaultWorkingDirectory)/**/.zip'
                    resourceGroupName: test

```

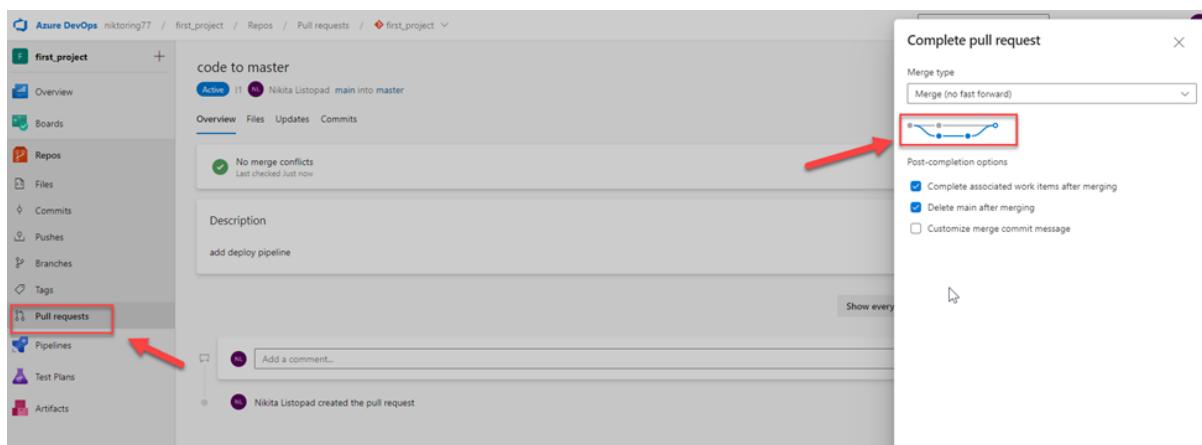
Do pipeline

The screenshot shows the Azure DevOps interface for the "first_project" repository. The left sidebar has "Pipelines" selected. A red box highlights the "Pipelines" button in the sidebar, and a red arrow points from it to the "Azure Repos Git" option in the "Where is your code?" section. The "Azure Repos Git" option is highlighted with a red box and has a hand cursor icon over it. Other options listed include "Bitbucket Cloud", "GitHub", "GitHub Enterprise Server", "Other Git", and "Subversion". Below the options, a note says "Use the classic editor to create a pipeline without YAML."

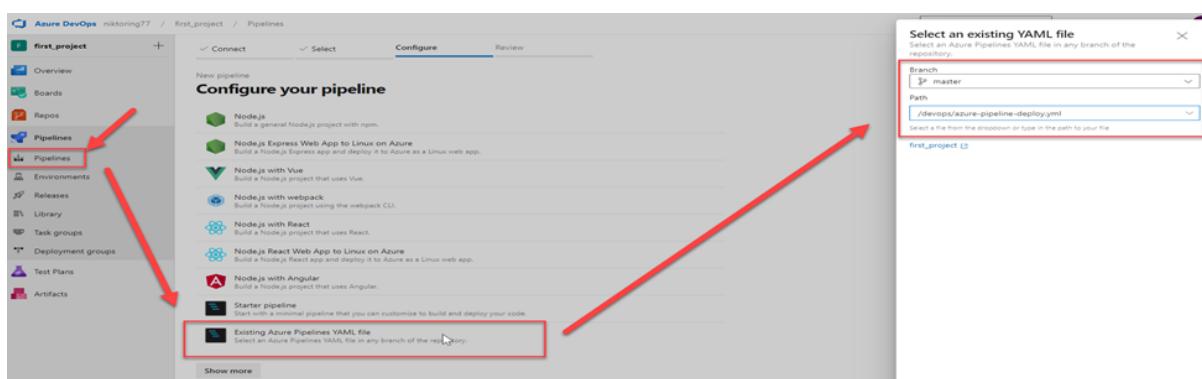
Choose our app



I pushed to main branch. I need create merge request to DEFAULT branch. in my case it is Master



Then



Continue

Azure DevOps niktoring77 / first_project / Pipelines

first_project

Overview Boards Repos Pipelines Environments Releases Library Task groups Deployment groups Test Plans Artifacts

New pipeline Review your pipeline YAML

```
first_project / devops/azure-pipeline-deployyaml #0
1 name: 'Validation running. Attempt ${builderCounter}'
2
3 trigger:
4   - main
5
6 variables:
7   - name: webAppName
8     value: 'webapp-epam'
9   - name: builderCounter
10    value: ${[counter('buildCounter', 1)]}
11
12 stages:
13   - stage: Build
14     displayName: 'Npm install/build'
15     jobs:
16       - deployment: "BuildApp"
17         pool:
18           vmImage: windows-2022
19           environment: validation
20           strategy:
21             runOnce:
22               deploy:
23                 steps:
24                   - checkout: self
25                   - clean: true
26                   - fetchDepth: 5
27
28             settings:
29               - task: Npm@1
30                 displayName: 'npm install'
31                 inputs:
32                   verbose: false
33
34             settings:
35               - task: Npm@1
36                 displayName: 'npm custom'
37                 inputs:
38                   command: custom
                    verbose: false
                    runCommand: 'run build'
```

Variables Run Save Show assistant

We have useful options

← first_project

Runs Branches Analytics

Get started and run this pipeline for the first time!

Run pipeline

Edit Run pipeline More

- Manage security
- Rename/move
- Status badge
- Trigger issues
- Settings
- Delete
- Scheduled runs

Edit

first_project / Pipelines / first_project

← first_project

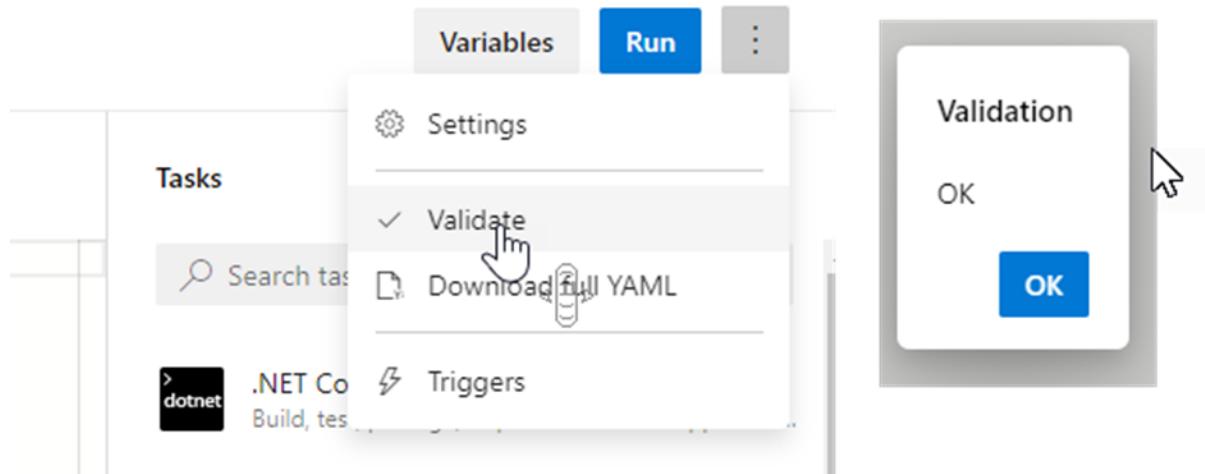
Runs Branches Analytics

Get started and run this pipeline for the first time!

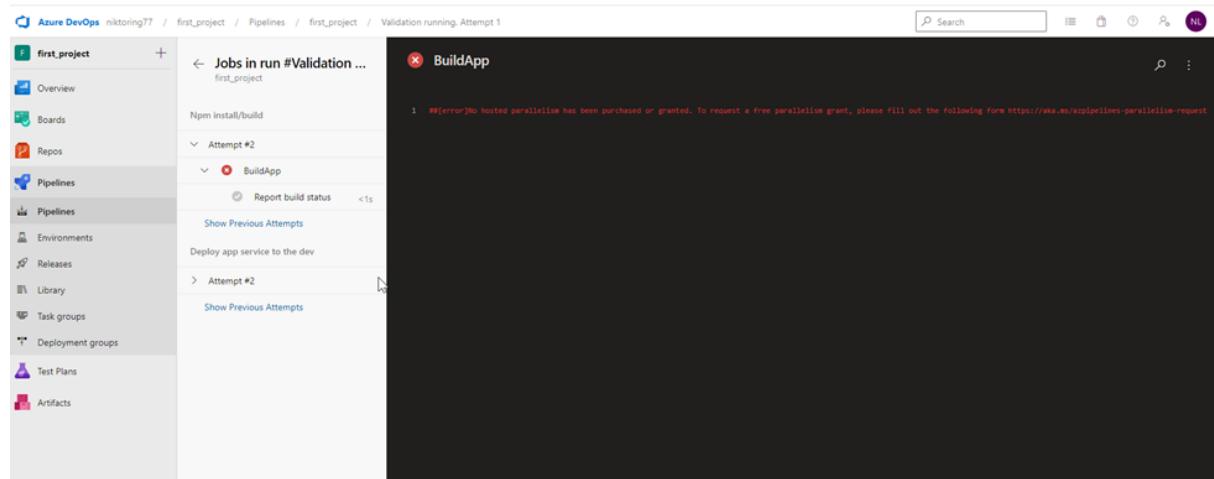
Run pipeline

Edit Run pipeline More

Check valid our file .yml



Run pipeline



Send request

<https://forms.office.com/pages/responsepage.aspx?id=v4j5cvGGr0GRqy180BHbR63mUWPlq7NEsFZhkyH8jChUMIM3QzdDMFZOMkVBWU5BWFM3SDI2QIRBSC4u>

Azure DevOps Parallelism Request

This form is for users to request increased parallelism in Azure DevOps.

Please consider that it could take 2-3 business days to proceed the request. We are working on improving this process at the moment. Sorry for the inconvenience.

...

* Required

1. What is your name? *

Enter your answer



2. What is your email address? *

Enter your answer



3. What is the name of your Azure DevOps Organization? *

(E.g. for <https://myorganization.visualstudio.com> or <https://dev.azure.com/myorganization> link formats - organization name would be 'myorganization')

Enter your answer

4. Are you requesting a parallelism increase for Public or Private projects? *

Private

Public

Submit

Never give out your password. [Report abuse](#)

Approval

Where is code on VM in Web-App

The screenshot shows the Microsoft Azure portal interface for a web application named "webApp-E pam". On the left, there is a sidebar with various navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Deployment (with Quickstart, Deployment slots, Deployment Center), and Settings (Configuration, Authentication). The main area is titled "webApp-E pam | Console" and shows a terminal window. The terminal output displays two directory listings:

```
D:\home\site\wwwroot>ls
README.md
angular.json
devops
dist
karma.conf.js
node_modules
package-lock.json
package.json
src
tsconfig.app.json
tsconfig.json
tsconfig.spec.json

D:\home\site\wwwroot>ls dist/todo
3rdpartylicenses.txt
assets
favicon.ico
index.html
main.d56d220b2719db70706f.js
polyfills.fa4c4cc7691890732dfd.js
runtime.16c801cf39e4720f0268.js
styles.bf8d580556def58427d3.css

D:\home\site\wwwroot>
```

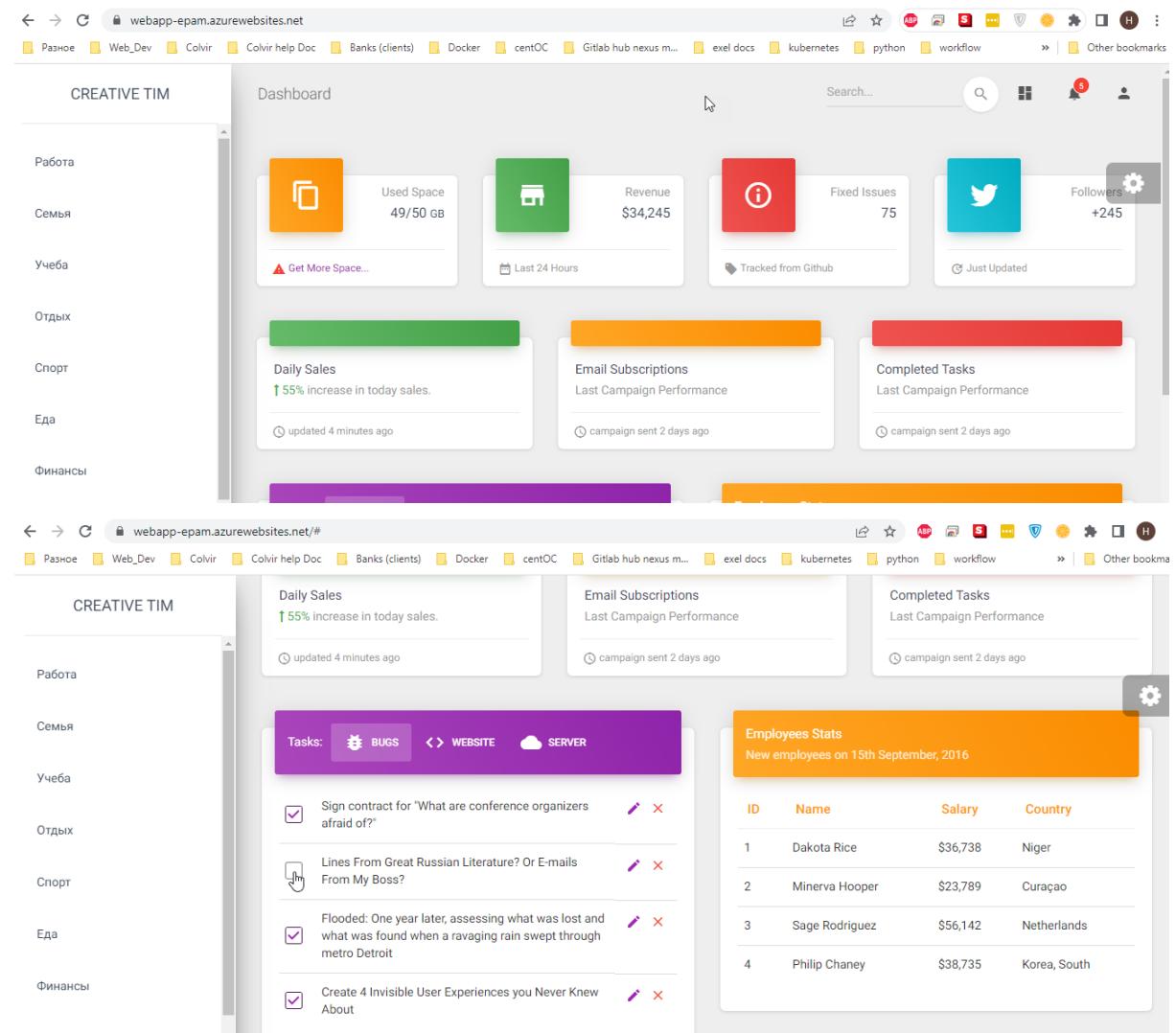
Successful pipeline after error bind with:

```
rootFolderOrFile: '$(System.DefaultWorkingDirectory)/dist/todo/*'
```

The screenshot shows the Azure DevOps interface for a project named "first_project". The left sidebar lists Pipelines, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Project settings. The main area shows a pipeline run for "Validation running, Attempt...". The pipeline steps listed are: Create an artifact (37s), Upload Artifacts (8s), Post-job: Checkout ... (<1s), Finalize Job (<1s), Deploy app service to the dev (1m 0s), Initialize job (7s), Download Artifact (11s), DownloadPipelineAr... (4s), Azure Web App De... (35s), and Finalize Job (<1s). The final step is "Report build status" (<1s). To the right, a detailed log for the "Azure Web App Deploy: webApp-E pam" step is shown:

```
1 Starting: Azure Web App Deploy: webApp-E pam
=====
3 Task      : Azure Web App
4 Description : Deploy an Azure Web App for Linux or Windows
5 Version   : 1.211.0
6 Author    : Microsoft Corporation
7 Help      : https://aka.ms/azureweapptroubleshooting
=====
9 Got service connection details for Azure App Service: 'webApp-E pam'
10 Trying to update App Service Application settings. Data: {"WEBSITE_RUN_FROM_PACKAGE":"1"}
11 Deleting App Service Application settings. Data: ["WEBSITE_RUN_FROM_ZIP"]
12 App Service Application settings are already present.
13 Package deployment using ZIP Deploy initiated.
14 Deploy logs can be viewed at https://webapp-epam.scm.azurewebsites.net/api/deployments/9f3c23he80a94fc9997de4
15 Successfully deployed web package to App Service.
16 NOTE: Run From Package makes wwwroot read-only, so you will receive an error when writing files to this directory.
17 Successfully added release annotation to the Application Insight : webApp-E pam
18 App Service Application URL: https://webapp-epam.azurewebsites.net
19 Finishing: Azure Web App Deploy: webApp-E pam
```

Show result



The screenshots show a web-based dashboard and task management system. The top part is a general dashboard with metrics like space usage, revenue, and follower counts. The bottom part is a specific task list with a focus on bugs, followed by a table of employee statistics.

ID	Name	Salary	Country
1	Dakota Rice	\$36,738	Niger
2	Minerva Hooper	\$23,789	Curaçao
3	Sage Rodriguez	\$56,142	Netherlands
4	Philip Chaney	\$38,735	Korea, South

Part 2 – Configure a pipeline to deploy infrastructure

Below is describing on how to do it via terraform. If you want to use terraform you need to create service connection in manual way. Otherwise you won't be able to deploy your iac – Navigate to the last section

Terraform storage account

1. **Create a separate resource group and deploy azure storage account -**
<https://learn.microsoft.com/en-us/azure/storage/common/storage-account-create?tabs=azure-portal>

Create storage account for some resource group!!!! Example for `epam_test`

Azure services

[Create a resource](#)

Cost Management ... Subscriptions App Services Azure DevOps organizations SQL databases Security

Resources

Recent Favorite

Name	Type
webApp-E pam	App Service
Azure_subscription_1	Subscription
webApp-E pam	Application Insights
epam_test	Resource group
nlistopadepam	App Service plan
server456	SQL server
DefaultResourceGroup-EUS	Resource group

See all

Microsoft Azure

Home > Create a resource > Marketplace > Storage account >

Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *

Resource group * [Create new](#)

Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name *

Region *

Performance Standard: Recommended for most scenarios (general-purpose v2 account)
 Premium: Recommended for scenarios that require low latency.

Redundancy *

[Review](#) [Next : Advanced >](#)

Create a storage account

...

Basics Advanced Networking Data protection Encryption Tags **Review**

Basics

Subscription	Azure_subscription_1
Resource Group	epam_test
Location	germanywestcentral
Storage account name	storageaccountforepam
Deployment model	Resource manager
Performance	Standard
Replication	Locally-redundant storage (LRS)

Advanced

Secure transfer	Enabled
Allow storage account key access	Enabled
Allow cross-tenant replication	Enabled
Default to Azure Active Directory authorization in the Azure portal	Disabled
Blob public access	Enabled
Minimum TLS version	Version 1.2 []
Permitted scope for copy operations (preview)	From any storage account
Enable hierarchical namespace	Disabled
Enable network file system v3	Disabled
Access tier	Hot
Enable SFTP	Disabled
Large file shares	Disabled

Networking

Network connectivity	Public endpoint (all networks)
Default routing tier	Microsoft network routing
Endpoint type	Standard

Data protection

Point-in-time restore	Disabled
Blob soft delete	Enabled
Blob retention period in days	7
Container soft delete	Enabled
Container retention period in days	7
File share soft delete	Enabled
File share retention period in days	7
Versioning	Disabled

Create



< Previous

Next >

Download a template for automation

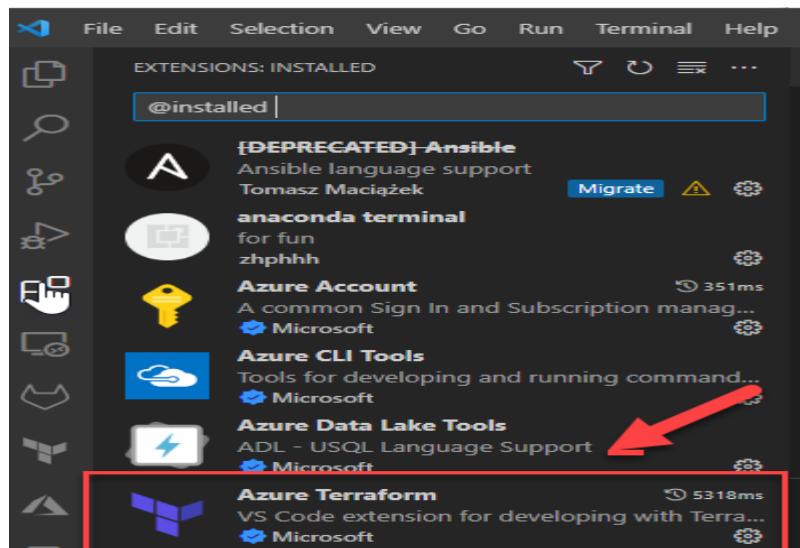
2. Create a container with the name “tfstate” and remember the name. use portal settings

The screenshot shows the Azure Storage account settings for 'storageaccountforepam'. The left sidebar includes links for Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, and Data storage. The 'Data storage' link is highlighted with a red arrow. The main content area displays essential details like Resource group, Location, Subscription, and Disk state. The 'Properties' tab is selected, showing blob service configurations such as Hierarchical namespace, Default access tier, Blob public access, Blob soft delete, and Container soft delete. Below this is the 'Containers' blade, which lists existing containers ('Slogs') and provides options to create a new container ('Container'). A 'New container' dialog is open on the right, prompting for a name ('tfstate') and public access level ('Private (no anonymous access)').

In this storage account you will be store your tf state file

Terraform preparation

prepare vsCode



The screenshot shows a code editor interface with a sidebar containing a file tree. The tree includes a folder named 'AZURE-REPO_TERRAFORM' which contains a 'terraform' folder, 'main.tf', 'outputs.tf', 'providers.tf', and 'variables.tf'. The 'main.tf' file is currently selected and highlighted with a blue background. The main pane displays the contents of 'main.tf':

```

8  # ...
9
10 # Configure the Azure provider
11 terraform {
12   required_providers {
13     azurerm = {
14       source  = "hashicorp/azurerm"
15       version = "~> 3.0.0"
16     }
17 }

```

Below the code editor is a terminal window showing the output of the 'terraform init' command:

```

PS E:\_Programming\_E-P-M\1\Epm-homeworks\04_Azure_homework\Azure-repo_terraform> terraform init
Terraform initialized in an empty directory!

```

The terminal also displays the welcome message for the Azure CLI and telemetry information:

```

The directory has no Terraform configuration files. You may begin working
with Terraform immediately by creating Terraform configuration files.
PS E:\_Programming\_E-P-M\1\Epm-homeworks\04_Azure_homework\Azure-repo_terraform> az

```

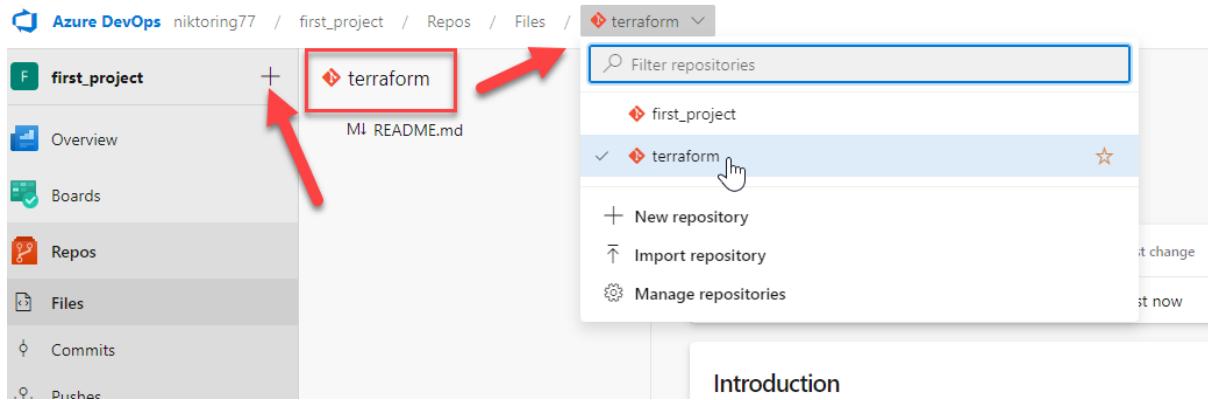
Below the terminal is another terminal window showing the output of the 'terraform -version' command:

```

PS E:\_Programming\_E-P-M\1\Epm-homeworks\04_Azure_homework\Azure-repo_terraform> terraform -version
Terraform v1.3.4
on windows_amd64
Your version of Terraform is out of date! The latest version
is 1.3.6. You can update by downloading from https://www.terraform.io/downloads.html
PS E:\_Programming\_E-P-M\1\Epm-homeworks\04_Azure_homework\Azure-repo_terraform>

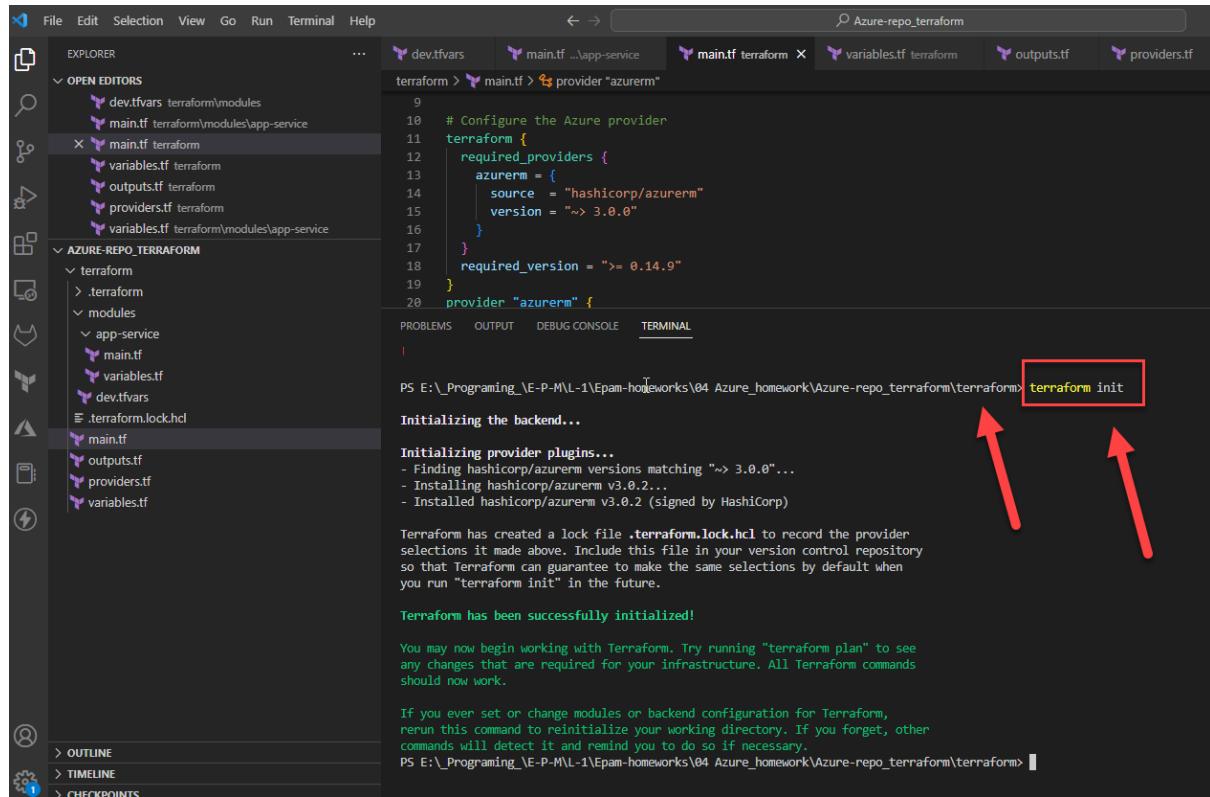
```

1. Create another repo to store devops code



2. Create a folder terraform

on local vsCode



The screenshot shows the VS Code interface with the terminal tab active. The command 'terraform init' is being run in the terminal. Red arrows point from the terminal output back up to the code editor, highlighting the configuration block being initialized.

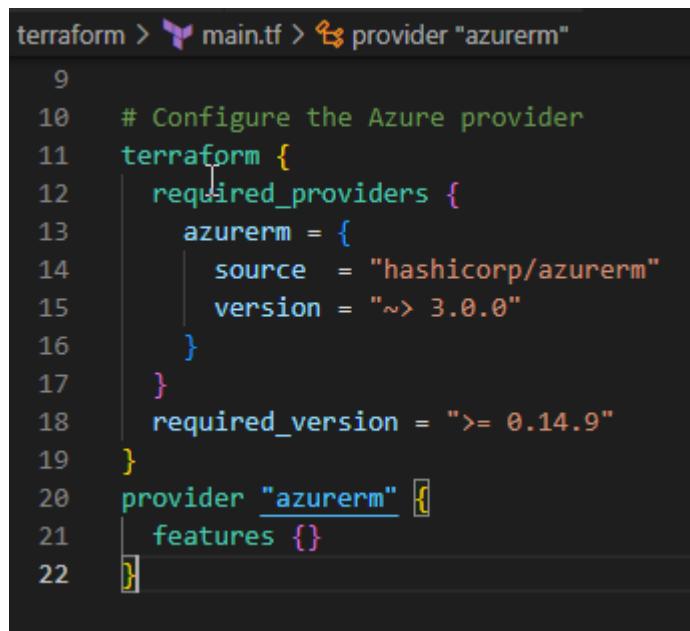
```
PS E:\_Programming\_E-P-M\1\Eepam-homeworks\04_Azure_homework\Azure-repo_terraform\terraform> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/azurerm versions matching "~> 3.0.0"...
- Installing hashicorp/azurerm v3.0.2...
- Installed hashicorp/azurerm v3.0.2 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.
PS E:\_Programming\_E-P-M\1\Eepam-homeworks\04_Azure_homework\Azure-repo_terraform\terraform>
```



The screenshot shows the code editor with the main.tf file open. The provider configuration block is highlighted with a red box. Red arrows point from the terminal output in the previous screenshot down to this highlighted code.

```
9
10 # Configure the Azure provider
11 terraform {
12     required_providers {
13         azurerm = {
14             source  = "hashicorp/azurerm"
15             version = "~> 3.0.0"
16         }
17     }
18     required_version = ">= 0.14.9"
19 }
20 provider "azurerm" {
21     features {}
22 }
```

cd terraform

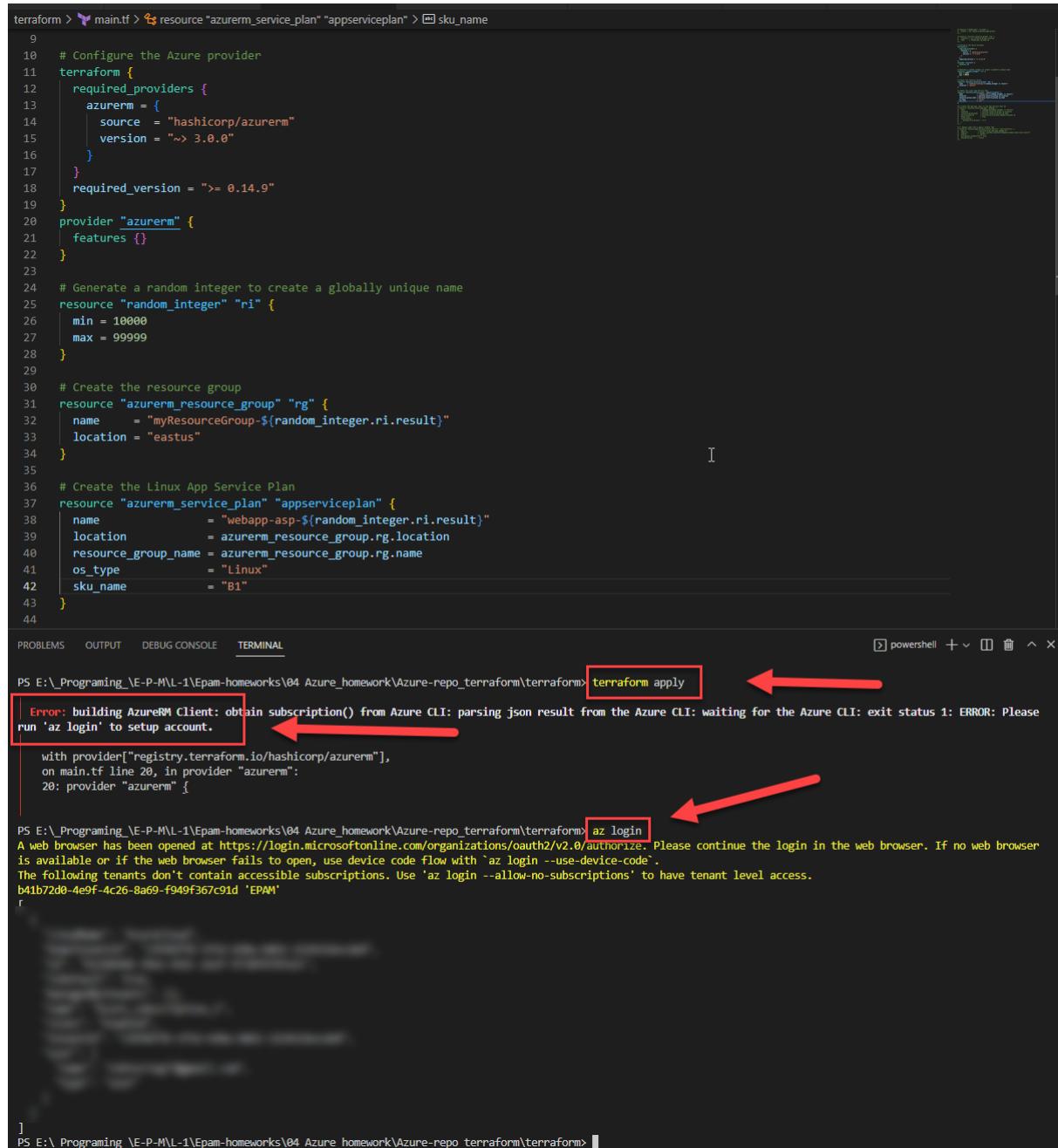
terraform init

3. Add app service implementation -

<https://learn.microsoft.com/en-us/azure/app-service/provision-resource-terraform>

az login because without login I can't apply and create resource and services!

connect to your existing Azure account



```
terraform > main.tf > resource "azurerm_service_plan" "appserviceplan" > sku_name
9
10 # Configure the Azure provider
11 terraform {
12   required_providers {
13     azurerm = {
14       source  = "hashicorp/azurerm"
15       version = "~> 3.0.0"
16     }
17   }
18   required_version = ">= 0.14.9"
19 }
20 provider "azurerm" {
21   features {}
22 }

23 # Generate a random integer to create a globally unique name
24 resource "random_integer" "ri" {
25   min = 10000
26   max = 99999
27 }
28 }

29 # Create the resource group
30 resource "azurerm_resource_group" "rg" {
31   name   = "myResourceGroup-${random_integer.ri.result}"
32   location = "eastus"
33 }

34 # Create the Linux App Service Plan
35 resource "azurerm_service_plan" "appserviceplan" {
36   name          = "webapp-asp-${random_integer.ri.result}"
37   location      = azurerm_resource_group.rg.location
38   resource_group_name = azurerm_resource_group.rg.name
39   os_type       = "Linux"
40   sku_name      = "B1"
41 }

42 }

43 }

44

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS E:\Programming\_E-P-M\1\Epm-homeworks\04_Azure_homework\Azure-repo_terraform> terraform apply
[ Error: building AzureRM Client: obtain subscription() from Azure CLI: parsing json result from the Azure CLI: waiting for the Azure CLI: exit status 1: ERROR: Please run 'az login' to setup account. ] ←
[ with provider["registry.terraform.io/hashicorp/azurerm"], on main.tf line 20, in provider "azurerm": ] ←
[ 20: provider "azurerm" { ] ←
[

PS E:\Programming\_E-P-M\1\Epm-homeworks\04_Azure_homework\Azure-repo_terraform> az login
A web browser has been opened at https://Login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with `az login --use-device-code`.
The following tenants don't contain accessible subscriptions. Use 'az login --allow-no-subscriptions' to have tenant level access.
b41b72d0-4e9f-4c26-8a69-f949f367c91d 'EPAM'
[

]

PS E:\Programming\_E-P-M\1\Epm-homeworks\04_Azure_homework\Azure-repo_terraform>
```

```

PS E:\_Programming_\E-P-M\1\Epm-homeworks\04 Azure_homework\Azure-repo_terraform\terraform> terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# azurerm_resource_group.rg will be created
+ resource "azurerm_resource_group" "rg" {
  + id      = (known after apply)
  + location = "eastus"
  + name    = (known after apply)
}

# azurerm_service_plan.appserviceplan will be created
+ resource "azurerm_service_plan" "appserviceplan" {
  + id          = (known after apply)
  + kind        = (known after apply)
  + location    = "eastus"
  + maximum_elastic_worker_count = (known after apply)
  + name        = (known after apply)
  + os_type     = "Linux"
  + per_site_scaling_enabled = false
  + reserved   = (known after apply)
  + resource_group_name = (known after apply)
  + sku_name    = "B1"
  + worker_count = (known after apply)
}

# random_integer.ri will be created
+ resource "random_integer" "ri" {
  + id      = (known after apply)
  + max    = 99999
  + min    = 10000
  + result = (known after apply)
}

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

random_integer.ri: Creating...
random_integer.ri: Creation complete after 0s [id=49393]
azurerm_resource_group.rg: Creating...
azurerm_resource_group.rg: Creation complete after 3s [id=/subscriptions/b124b468-39ee-416c-a1ef-9738f476fa3c/resourceGroups/myResourceGroup-49393]
azurerm_service_plan.appserviceplan: Creating...
azurerm_service_plan.appserviceplan: Still creating... [10s elapsed]
azurerm_service_plan.appserviceplan: Still creating... [20s elapsed]
azurerm_service_plan.appserviceplan: Creation complete after 23s [id=/subscriptions/b124b468-39ee-416c-a1ef-9738f476fa3c/resourceGroups/myResourceGroup-49393/providers/Microsoft.Web/serverfarms/webapp-asp-49393]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
PS E:\_Programming_\E-P-M\1\Epm-homeworks\04 Azure_homework\Azure-repo_terraform\terraform>

```

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes 'Microsoft Azure', 'Search resources, services, and docs (G+)', and a user profile icon. Below the navigation bar, the URL 'PS E:_Programming_\E-P-M\1\Epm-homeworks\04 Azure_homework\Azure-repo_terraform\terraform>' is visible.

The main content area displays the 'Resource groups' page for the 'myResourceGroup-49393' group. The left sidebar lists other resource groups: 'cloud-shell-storage-westeurope', 'DefaultResourceGroup-EUS', 'epam_test', and 'myResourceGroup-49393'. A red arrow points to the 'myResourceGroup-49393' entry. The main panel shows the 'Overview' tab for the selected group. It displays basic information: Subscription (move) : Azure_subscription_1, Subscription ID : b124b468-39ee-416c-a1ef-9738f476fa3c, Tags (edit) : Click here to add tags, Deployments : No deployments, and Location : East US. The 'Resources' tab is active, showing one record: 'webapp-asp-49393'. A red arrow points to this entry. The bottom of the screen shows the Windows taskbar with icons for File Explorer, Task View, Start, and Task Manager.

4. Integrate application insights with app service

Home > myResourceGroup-49393

myResourceGroup-49393 | Insights (preview)

Resource group

Search Refresh Collapse all Feedback Help

Overview Activity log Access control (IAM) Tags Resource visualizer Events

Settings Deployments Security Policies Properties Locks

Cost Management Cost analysis Cost alerts (preview) Budgets Advisor recommendations

Monitoring Insights (preview) Alerts Metrics

Total resources Active alerts 1

Filter by name...

NAME	TOTAL ALERTS	SEV 0 ALERTS	SEV 1 ALERTS	INSIGHTS
myResourceGroup-49393	—	—	—	
Compute	—	—	—	
App Service plan	—	—	—	
webapp-asp-49393	—	—	—	

/subscriptions/b124b468-39ee-416c-a1ef-9738f476fa3c/resourceGroups/myResourceGroup-49393/providers/Microsoft.Web/serverFarms/webapp-asp-49393

5. Updated backend "azurerm" according to the guide -

<https://learn.microsoft.com/en-us/azure/developer/terraform/store-state-in-azure-storage?tabs=azure-cli>

devitvars manifl ...app-service terraform tfstate variables.tf outputs.tf

backend "azurerm" {
 resource_group_name = "myResourceGroup-49393"
 storage_account_name = "storegaccfortest"
 container_name = "tfstate"
 key = "terraform.tfstate"
}

PS E:\Programming\VS-P-MU\1Epam-homeworks\04_Azure_homework\Azure-repo_terraform> terraform init --reconfigure

Initializing the backend...
Do you want to copy existing state to the new backend?
Pre-existing state was found while migrating the previous "local" backend to the newly configured "azurerm" backend. No existing state was found in the newly configured "azurerm" backend. Do you want to copy this state to the new "azurerm" backend? Enter "yes" to copy and "no" to start with an empty state.

Enter a value: yes

Successfully configured the backend "azurerm"! Terraform will automatically use this backend unless the backend configuration changes.

Initializing provider plugins...
- Reusing previous version of hashicorp/azurerm from the dependency lock file
- Reusing previous version of hashicorp/random from the dependency lock file
- Using previously-installed hashicorp/azurerm v3.0.2
- Using previously-installed hashicorp/random v3.4.3

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, run this command to reinitialize your working directory. If you forget, other commands will detect it and prompt you to do this if necessary.

PS E:\Programming\VS-P-MU\1Epam-homeworks\04_Azure_homework\Azure-repo_terraform>

portal.azure.com / Microsoft Azure

Create a resource Resource groups Storage accounts Cost Management ... Subsci

Resources

myResourceGroup-49393 storegaccfortest webapp-asp-49393 epam_test Azure_subscription_1 webApp-Epac webApp-Epac nistopadepam server456 DefaultResourceGroup-EUS

Navigate

Subscriptions Resource groups

Tools

Microsoft Learn Microsoft Learn Azure with free online training from Microsoft Azure Monitor Monitor your apps and infrastructure

The screenshot shows two main sections of the Microsoft Azure portal:

- Resource Group Overview:** On the left, under the 'Resources' tab, a new resource named 'storageaccfortest' is listed. A red arrow points to this resource.
- Storage Account Containers:** On the right, in the 'Containers' blade, a new container named 'tfstate' is being created. A red arrow points to the 'tfstate' container.

Below these, a detailed view of the 'tfstate' blob container is shown, displaying the contents of the 'terraform.tfstate' file. The file contains Terraform state data for an Azure Resource Group, including resources like 'azurerm_resource_group' and 'azurerm_service_plan'.

```

tfstate
+-- terraform.tfstate
  +-- terraform.tfstate
    +-- schema_version: 1
    +-- serial: 14
    +-- lineage: "0e691b41-9be0-2b16-cc4f-2f5c06823e9b"
    +-- outputs: {}
    +-- resources: [
        +-- resource_group: {
            +-- mode: "managed"
            +-- type: "azurerm_resource_group"
            +-- name: "rg"
            +-- provider: "provider[\"registry.terraform.io/hashicorp/azurerm\"]"
            +-- instances: [
                +-- id: "/subscriptions/b124b468-39ee-416c-a1ef-9738f476fa3c/resourceGroups/myResourceGroup-49393"
                +-- location: "eastus"
                +-- name: "myResourceGroup-49393"
                +-- tags: null
                +-- timeouts: null
            ]
            +-- sensitive_attributes: []
            +-- provider: "provider[\"registry.terraform.io/hashicorp/azurerm\"]"
            +-- dependencies: []
            +-- random_integer: null
        }
        +-- appserviceplan: {
            +-- mode: "managed"
            +-- type: "azurerm_service_plan"
            +-- name: "appserviceplan"
            +-- provider: "provider[\"registry.terraform.io/hashicorp/azurerm\"]"
            +-- instances: [
                +-- id: "71411403-3311-4211-9200-000000000000"
                +-- name: "appserviceplan"
                +-- schema_version: 0
            ]
        }
    ]
  ]
}
  
```

6. Run az login or Connect-AzAccount to connect the azure subscription from your local

az login

connect to your existing Azure account

The screenshot shows a terminal window and a Microsoft Azure sign-in dialog box side-by-side.

Terminal Window:

```

PS E:\Programming\EP-MVL-1\EPM-homeworks\04_Azure_homework\Azure-repo_terraform> az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize?... Please
continue the login in the web browser. If no web browser is available or if the web browser fails to open, use
device code flow with 'az login --use-device-code'.
  
```

Microsoft Azure Sign-in Dialog:

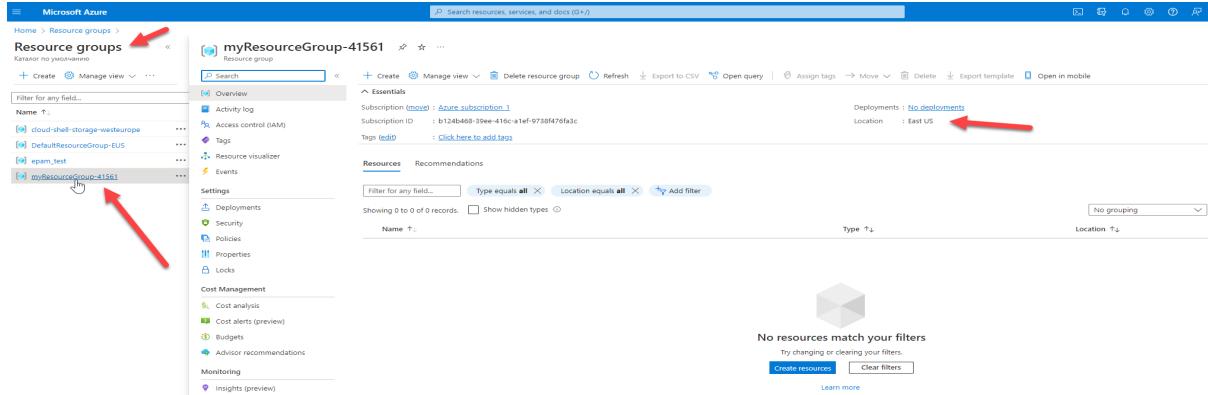
The dialog box displays the message: "Pick an account". It shows a user profile for "Mykyta Listopad" with the email "niktoring77@gmail.com" and the status "Signed in". There is also a "Use another account" button and a "Back" button.

7. Run terraform apply to deploy infrastructure

Important note: Use only freshest version of tf module like

https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/windows_web_app

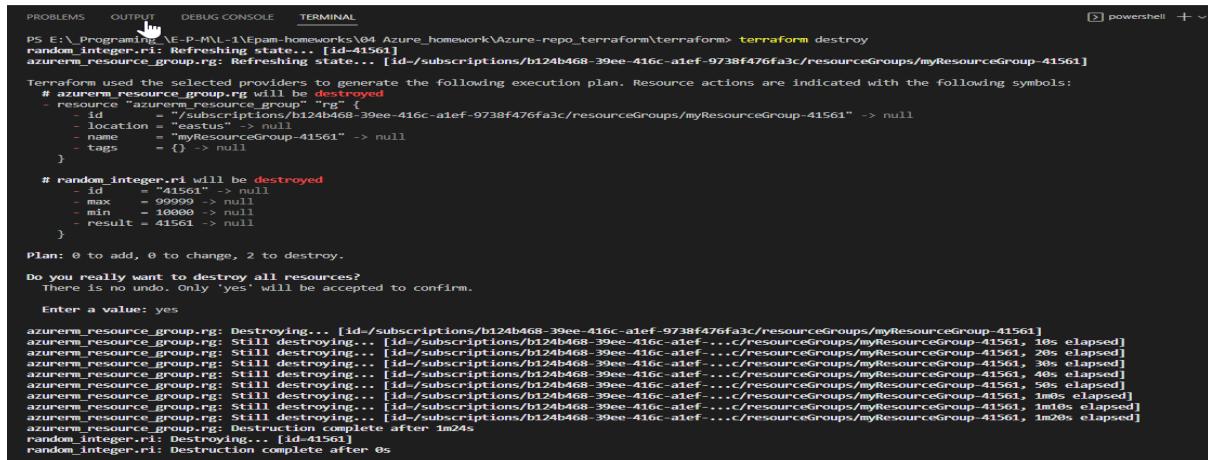
terraform apply create resource group in Azure cloud!



The screenshot shows the Microsoft Azure portal's Resource groups page. In the left sidebar, there is a list of existing resource groups: 'cloud-shell-storage-westeurope', 'DefaultResourceGroup-EUS', 'epam_test', and 'myResourceGroup-41561'. A red arrow points to the 'myResourceGroup-41561' entry. At the top right of the main content area, there is a 'Location' dropdown set to 'East US', with another red arrow pointing to it. The central area displays a message: 'No resources match your filters'.

terraform destroy delete this group from cloud

Important note: Don't forget to destroy your application once completed



```
PS E:\Programs\...\random_integer.tf: Refreshing state... [id=41561]
azurerm_resource_group.rg: Refreshing state... [id=/subscriptions/b124b468-39ee-416c-a1ef-9738f476fa3c/resourceGroups/myResourceGroup-41561]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
# azurerm_resource_group.rg will be destroyed
resource "azurerm_resource_group" "rg" {
  - id      = "/subscriptions/b124b468-39ee-416c-a1ef-9738f476fa3c/resourceGroups/myResourceGroup-41561" -> null
  - location = "eastus" -> null
  - name    = "myResourceGroup-41561" -> null
  - tags    = {} -> null
}

# random_integer.rg will be destroyed
resource "random_integer" "rg" {
  - id      = "41561" -> null
  - max    = 99999 -> null
  - min    = 10000 -> null
  - result = 41561 -> null
}

Plan: 0 to add, 0 to change, 2 to destroy.

Do you really want to destroy all resources?
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

azurerm_resource_group.rg: Destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-9738f476fa3c/resourceGroups/myResourceGroup-41561]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-.../resourceGroups/myResourceGroup-41561, 10s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-.../resourceGroups/myResourceGroup-41561, 20s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-.../resourceGroups/myResourceGroup-41561, 30s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-.../resourceGroups/myResourceGroup-41561, 40s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-.../resourceGroups/myResourceGroup-41561, 50s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-.../resourceGroups/myResourceGroup-41561, 1m0s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-.../resourceGroups/myResourceGroup-41561, 1m10s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b124b468-39ee-416c-a1ef-.../resourceGroups/myResourceGroup-41561, 1m20s elapsed]
azurerm_resource_group.rg: Destruction complete after 1m24s
random_integer.rg: Destroying... [id=41561]
random_integer.rg: Destruction complete after 0s
```

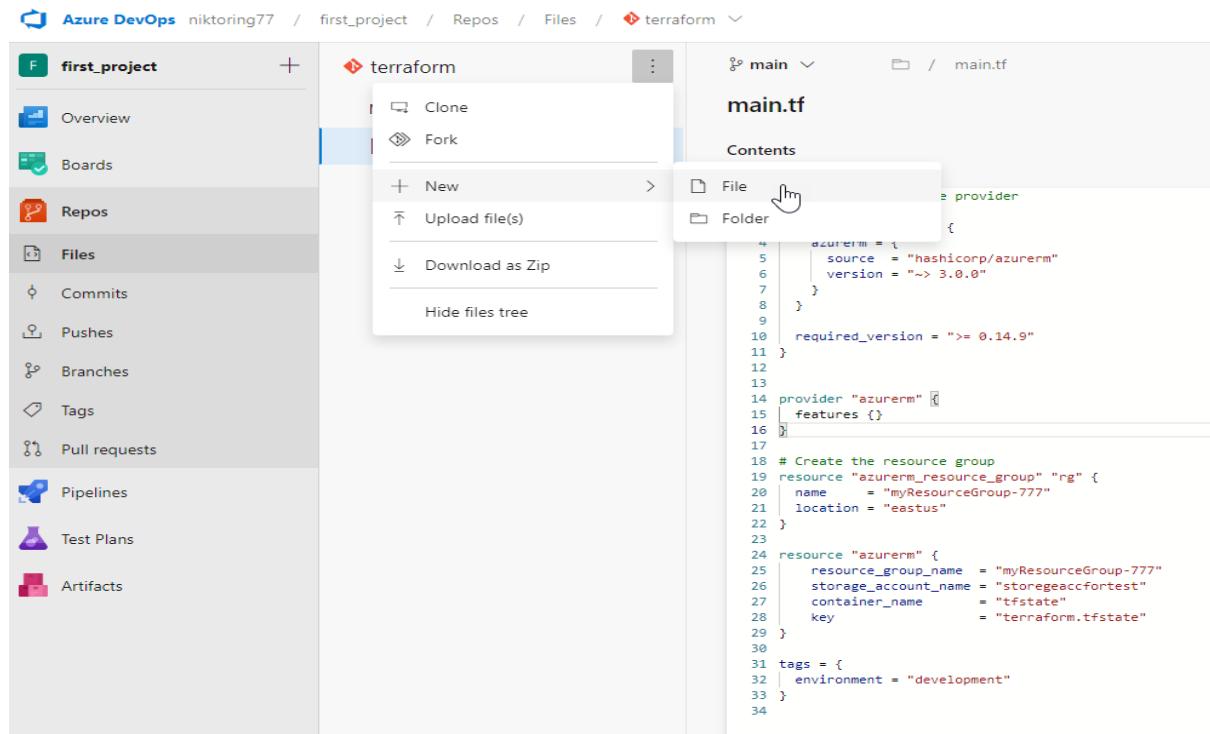
Create a terraform pipeline

1. Create a yaml pipeline with the following steps: terraform install, terraform init, terraform plan/apply. Plan is an optional one
2. Inside yaml pipeline add trigger to main branch. The scenario – when main is updated, pipeline should run automatically -
<https://learn.microsoft.com/en-us/azure/devops/pipelines/yaml-schema/trigger?view=azure-pipelines>

3. Added 3 steps – terraform install, terraform init, terraform plan/apply. Plan is an optional one.
 You may add it as 4th step

MANUIL

Create repo for terraform:



```

provider "azurerm" {
  source  = "hashicorp/azurerm"
  version = "~> 3.0.0"
}

provider "azurerm" {
  features {}
}

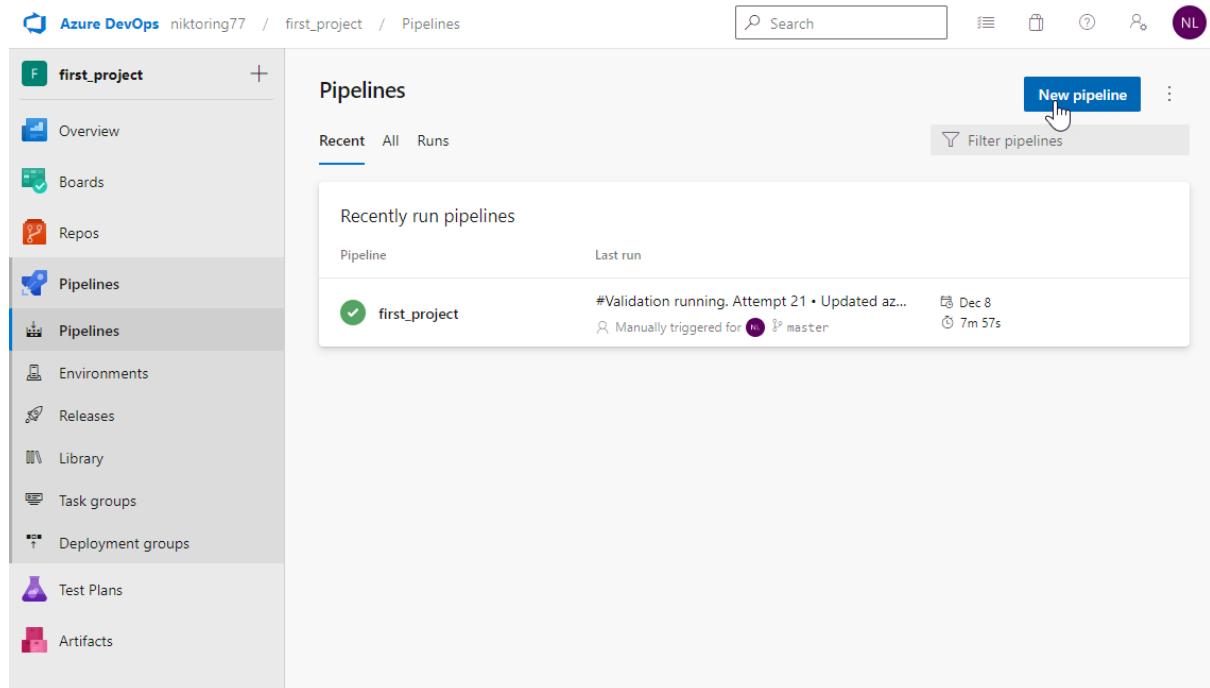
# Create the resource group
resource "azurerm_resource_group" "rg" {
  name     = "myResourceGroup-777"
  location = "eastus"
}

resource "azurerm" {
  resource_group_name = "myResourceGroup-777"
  storage_account_name = "storageaccfortest"
  container_name       = "tfstate"
  key                 = "terraform.tfstate"
}

tags = {
  environment = "development"
}

```

Create pipeline for artifact:



Pipeline	Last run
first_project	#Validation running. Attempt 21 • Updated az... Manually triggered for master Dec 8 7m 57s

Another way to create an artifact for creating infrastructure and write state in existing (maybe some default group) group and storage account

```

# Configure the Azure provider
provider "azurerm" {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "> 3.0.0"
    }
  }
}

backend "azurerm" {
  resource_group_name = var.resource_group_name
  storage_account_name = var.storage_account_name_storage
  container_name        = "tfstate"
  key                  = "terraform.tfstate"
}

provider "azurerm" {
  features {
    resource_group {
      prevent_deletion_if_contains_resources = false
    }
  }
}

# Create the resource group
resource "azurerm_resource_group" "rg" {
  name     = var.resource_group_name
  location = var.resource_group_location
}

# Create the app-service-plan
resource "azurerm_app_service_plan" "app" {
  name                = "appserviceplan"
  location            = var.resource_group_location
  resource_group_name = var.resource_group_name
  sku {
    tier = "Free"
    size = "F1"
  }
}

# Create the app-service-plan
resource "azurerm_app_service" "dev" {
  name                = "appservicev1name-876555"
  location            = var.resource_group_location
  resource_group_name = var.resource_group_name
  app_service_plan_id = azurerm_app_service_plan.app.id
}

```

```

# project vars
variable "resource_group_location" {
  default = "North Europe"
  description = "location of the resource group."
}

variable "resource_group_name" {
  default = "myResourceGroup-777"
  description = "Resource group name"
}

variable "storage_account_name" {
  default = "storageaccountlist"
  description = "Storage account name"
}

variable "resource_group_name_storage" {
  default = "cloud-shell-storage-westeuropa"
  description = "Resource group name"
}

variable "storage_account_name_storage" {
  default = "csb003200249e95bdc"
  description = "Storage account name"
}

```

Use the classic editor to create a pipeline without YAML.

Select a source

- Azure Repos Git
- Github
- GitHub Enterprise Server
- Subversion
- Bitbucket Cloud
- Other Git

Team project

Repository

Default branch for manual and scheduled builds

Continue

first_project

Pipelines

Tasks Variables Triggers Options History Save & queue Discard Summary Queue ...

Get sources

Agent job 1

+

Add tasks Refresh copy

Copy files over SSH

Copy files

Azure file copy

Windows machine file copy

Empty job

... > first_project-CI

Tasks Variables Triggers Options History | Save & queue Discard Summary Queue ...

Pipeline

Get sources

Agent job 1

Run on agent

Copy Files to:

Some settings need attention

Add tasks Refresh

publ|

Publish Test Results

Publish test results to Azure Pipelines

Publish build artifacts

Publish build artifacts to Azure Pipelines or a Windows file share

.NET Core

Add

Azure Pipelines CI pipeline interface. The pipeline consists of:

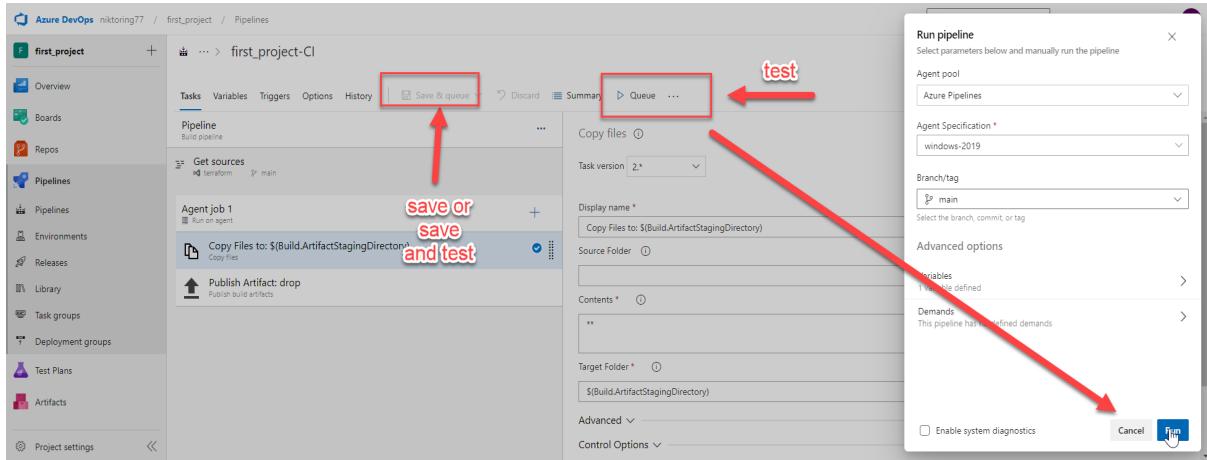
- Get sources (using terraform, main branch)
- Agent job 1 (Run on agent)
- Copy Files to: \$(Build.ArtifactStagingDirectory)
- Publish Artifact: drop (selected, highlighted in blue)
 - Task version: 1.*
 - Display name: Publish Artifact: drop
 - Path to publish: \$(Build.ArtifactStagingDirectory)
 - Artifact name: drop
 - Artifact publish location: Azure Pipelines

The 'Publish Artifact: drop' task has a red arrow pointing to its 'Path to publish' field, which contains the value '\$(Build.ArtifactStagingDirectory)'.

The screenshot shows the Azure DevOps Pipeline Editor. On the left, a pipeline named 'first_project-Cl' is displayed with three stages: 'Get sources', 'Agent job 1', and 'Copy Files to: \$(Build.ArtifactStagingDirectory)'. The third stage is currently selected. On the right, the 'Copy files' task configuration is shown with the following details:

- Task version:** 2.*
- Display name:** Copy Files to: \$(Build.ArtifactStagingDirectory)
- Source Folder:** (empty)
- Contents:** **
- Target Folder:** \$(Build.ArtifactStagingDirectory)

Red arrows point from the 'Source Folder' field in the pipeline to the 'Source Folder' field in the task configuration, and from the 'Target Folder' field in the pipeline to the 'Target Folder' field in the task configuration. The word 'auto' is highlighted in red under the 'Contents' field.



#24 • Added main.tf

Summary

Manually run by Nikita Listopad

Repository and version

terraform

main e1b4360d

Jobs

Name

Agent job 1

Jobs in run #24

Agent job 1

1 Pool: Azure Pipelines
2 Image: windows-2019
3 Agent: Hosted Agent
4 Started: Just now
5 Duration: 12s
6
7 ► Job preparation parameters
8 ► 1 queue time variable used
9 □ 1 artifact produced

Artifacts

Published Consumed

Name

drop .git main.tf README.md

Pipelines

- Pipelines
- Environments
- Releases** (highlighted with a red arrow)
- Library
- Task groups
- Deployment groups
- Test Plans

No release pipelines found

Create your release process in a few easy steps with a new pipeline

New pipeline

Select a template
Or start with an **Empty job**

Featured

- Azure App Service: Deploy your application to Web App on Windows, Linux, or WebJobs.
- Deploy a Java app: Deploy a Java application.
- Deploy a Node.js app: Deploy a Node.js application.
- Deploy a PHP app: Azure Database for MySQL - Deploy a PHP application.

Save **Create release**

All pipelines > New release pipeline

Pipeline **Tasks** **Variables** **Retention** **Options** **History**

Artifacts | + Add **Stages | + Add**

+ Add an artifact **Stage 1** Select a template

Schedule not set

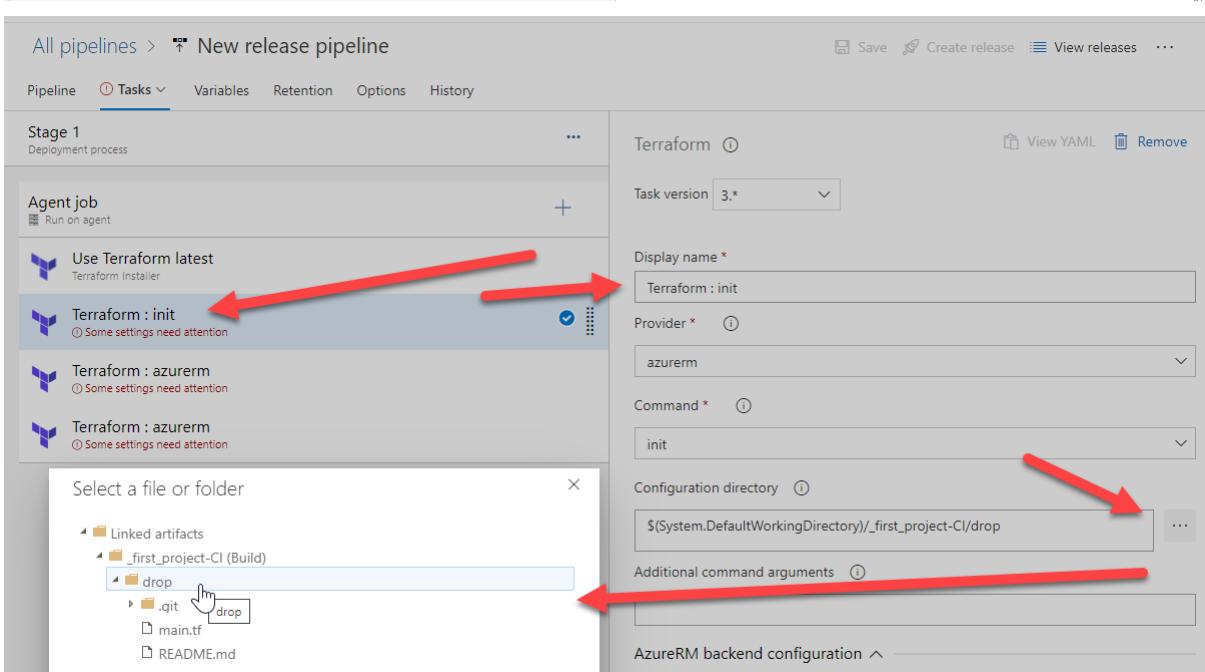
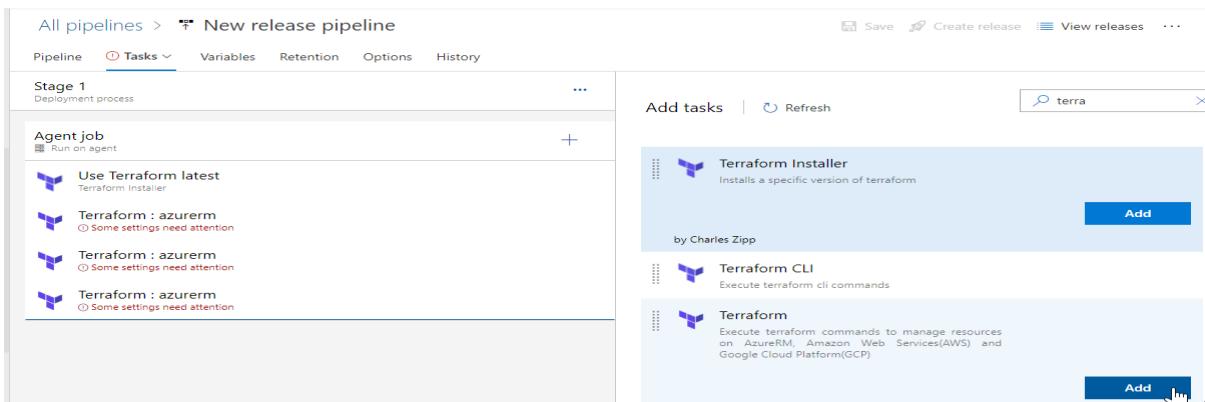
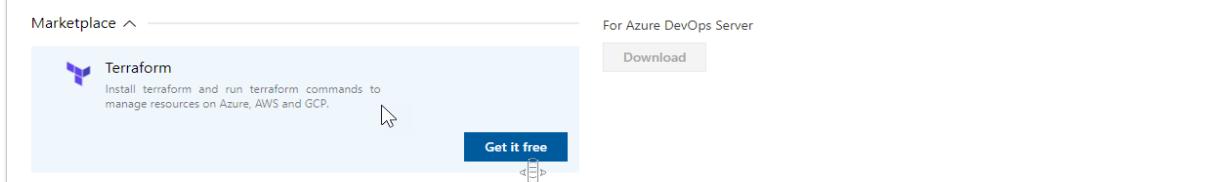
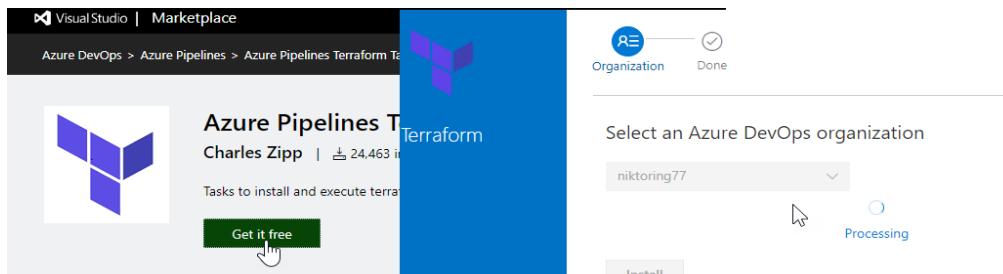
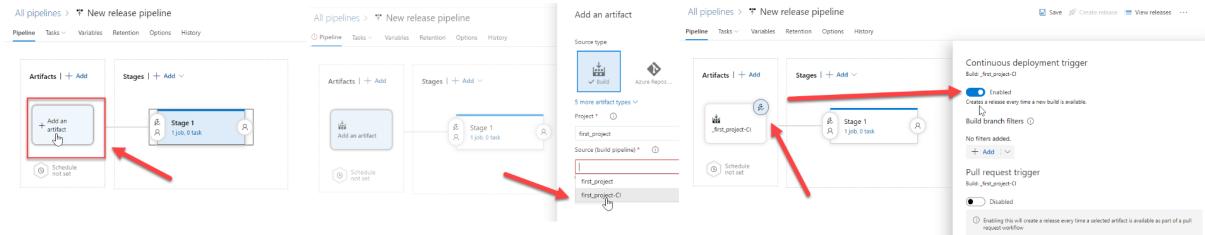
Stage
Stage 1

Properties Name and owners of the stage
Stage name Stage 1
Stage owner Nikita Listopad

Continuous deployment trigger
Build: _first_project-CI
Enabled (switch is blue)
Creates a release every time a new build is available.
Build branch filters (1)
No filters added.
+ Add | ↴

Pull request trigger
Build: _first_project-CI
Disabled
Enabling this will create a new release every time a selected artifact is available as part of a pull request workflow

Enabling the trigger will create a new release every time a new build is available.



All pipelines > **New release pipeline**

Pipeline Tasks Variables Retention Options History

Stage 1 Deployment process

Agent job Run on agent

- Use Terraform latest Terraform Installer
- Terraform : init Some settings need attention
- Terraform : azurerm Some settings need attention
- Terraform : azurerm Some settings need attention

Terraform View YAML Remove

Task version 3.*

Display name * Terraform : init

Provider * azurerm

Command * init

Configuration directory ... \$(System.DefaultWorkingDirectory)/_first_project-CI/drop

Additional command arguments

AzureRM backend configuration ^

Azure subscription * My Subscription Manage

Resource group * myResourceGroup-49393

Storage account * storegeaccfortest

Container * tfstate

Key * (i) The path to the Terraform remote state file inside the container. For example, if you want to store the state file, named terraform.tfstate, inside a folder, named tf, then give the input tf/terraform.tfstate

This setting is required.

Control Options

Output Variables

All pipelines > **New release pipeline**

Pipeline Tasks Variables Retention Options History

Stage 1 Deployment process

Agent job Run on agent

- Use Terraform latest Terraform Installer
- Terraform : init Terraform
- Terraform : plan** Terraform
- Terraform : azurerm Some settings need attention

Terraform View YAML Remove

Task version 3.*

Display name * Terraform : plan

Provider * azurerm

Command * plan

Configuration directory ... \$(System.DefaultWorkingDirectory)/_first_project-CI/drop

Additional command arguments

Azure subscription * My Subscription Manage

Control Options

Output Variables

Option 1

The screenshot shows the Azure DevOps Pipeline Editor for a "New release pipeline". The pipeline consists of two stages: Stage 1 (Deployment process) and Stage 2 (Deployment process). Stage 1 contains four tasks: "Agent job" (Run on agent), "Use Terraform latest" (Terraform Installer), "Terraform : init" (Terraform), and "Terraform : azurerm" (Terraform). Stage 2 contains three tasks: "Agent job" (Run on agent), "Terraform : azurerm" (Terraform), and "Terraform : azurerm" (Terraform). A red box highlights the Terraform configuration code in Stage 1:

```

7 }
8 }
9 required_version = ">= 0.14.9"
10 }
11 }
12 provider "azurerm" {
13   features {}
14 }
15 # Create the resource group
16 resource "azurerm_resource_group" "rg" {
17   name       = "myResourceGroup-777"
18   location   = "eastus"
19 }
20
21 resource "azurerm" {
22   resource_group_name = "myResourceGroup-777"
23   storage_account_name = "storageaccfortest"
24   container_name       = "tfstate"
25   key                  = "terraform.tfstate"
26 }
27
28 tags = {
29   environment = "development"
30 }
31
32 }
33
34
  
```

Red arrows point from the highlighted code to the "Resource group", "Storage account", and "Container" fields in the "Terraform : azurerm" task in Stage 2. The "Container" field is set to "tfstate". The "Key" field is set to "terraform.tfstate". A tooltip for the "Key" field explains: "The path to the Terraform remote state file inside the container. For example, if you want to store the state file, named terraform.tfstate, inside a folder, named tf, then give the input tf/terraform.tfstate".

Option 2

The screenshot shows the Azure DevOps Pipeline Editor for a "create_infrastructure" pipeline. The pipeline consists of four stages: Stage 1 (Deployment process), Stage 2 (Deployment process), Stage 3 (Deployment process), and Stage 4 (Deployment process). Stage 1 contains three tasks: "Agent job" (Run on agent), "Install Terraform latest" (Terraform tool installer), and "Terraform : init" (Terraform). Stage 2 contains three tasks: "Terraform : plan" (Terraform) and "Terraform : apply" (Terraform). Stage 3 contains one task: "Terraform : plan" (Terraform). Stage 4 contains one task: "Terraform : apply" (Terraform). A red box highlights the Terraform configuration code in Stage 3:

Have already existed some default

Resource group
and Storage account in it
for state file

Red arrows point from the highlighted text to the "Resource group", "Storage account", and "Container" fields in the "Terraform : plan" task in Stage 4. The "Resource group" is set to "cloud-shell-storage-westeuropa", the "Storage account" is set to "csb100320024a9a95bc", and the "Container" is set to "tfstate". The "Key" field is set to "terraform.tfstate".

The screenshot shows the Azure DevOps Pipeline Editor. On the left, under 'Stage 1 Deployment process', there are four Terraform tasks: 'Use Terraform latest', 'Terraform : init', 'Terraform : plan', and 'Terraform : apply'. Red arrows point from the 'Terraform : plan' and 'Terraform : apply' tasks to their corresponding configuration fields in the right panel. The right panel is titled 'Terraform' and contains the following fields:

- Task version:** 3.x
- Display name:** Terraform : apply
- Provider:** azurerm
- Command:** apply
- Configuration directory:** \${System.DefaultWorkingDirectory}/_first_project-CI/drop
- Additional command arguments:** -auto-approve
- Azure subscription:** My_Subscription
- Control Options:** (empty)
- Output Variables:** (empty)

SAVE

The screenshot shows the Azure DevOps interface during a pipeline run. On the left, the project navigation bar includes 'Pipelines'. In the center, the pipeline status is shown as 'In progress'. The pipeline has three stages: Stage 1, Stage 2, and Stage 3. Stage 1 is currently active, showing an 'Agent job' step. The logs for this step show the execution of Terraform commands. Stage 2 and Stage 3 are also listed as completed ('Succeeded'). The right side of the screen shows the detailed logs for each stage.

when I fix, add, rewrite some to file **main.tf** and create **ARTIFACT**

The screenshot shows the Azure DevOps interface. On the left, the project navigation bar includes 'Pipelines'. In the center, the 'first_project' repository is open, showing the 'main.tf' file. A red arrow points from the 'Edit' button in the main.tf file to the 'first.project-CI' pipeline history on the right. The pipeline history shows three recent commits:

- #26 - Updated main.tf (Manually triggered by main) - 4m ago
- #25 - Updated main.tf (Manually triggered by main) - 17m ago
- #24 - Added main.tf (Manually triggered by main) - 1h ago

RELEASE start automatically

Pipeline with .yaml file

Add the terraform extension from marketplace first of all

```

# Starter pipeline
# Start with a minimal pipeline that you can customize to build and deploy your code.
# Add steps for build, test, deployment, etc.
# For more information about the workflow tasks, see: https://aka.ms/yaml

trigger: none

pool:
  vmImage: ubuntu-latest

variables:
  resource_group_name: "MyResourceGroup-777"
  storage_account_name: "MyStorageAccount"
  blob_end_container_name: "tfstate"
  blob_and_telemetry_key: "terraformstate"
  tfstate_file: ".tfstate"

stages:
  - stage: tfvalidate
    jobs:
      - job: validate
        continuedOnError: false
        steps:
          - task: TerraformInstaller@0
            inputs:
              provider: "azurerm"
              command: "list"
            dependsOn:
              - step: validate
            condition: succeeded('tfvalidate')
            strategy:
              evaluate: true
              validate: true
            steps:
              - task: TerraformValidate@0
                displayName: tfvalidate
                inputs:
                  provider: "azurerm"
                  command: "validate"
            dependsOn:
              - stage: tfdeploy
            condition: succeeded('tfvalidate')
            strategy:
              evaluate: true
              validate: true
            steps:
              - task: TerraformDeploy@0
                displayName: tfdeploy
                inputs:
                  provider: "azurerm"
                  command: "apply"
            dependsOn:
              - stage: tfvalidate
            condition: succeeded('tfvalidate')
            strategy:
              evaluate: true
              validate: true
            steps:
              - task: TerraformDestroy@0
                displayName: tfdestroy
                inputs:
                  provider: "azurerm"
                  command: "destroy"
            dependsOn:
              - stage: tfdeploy
            condition: succeeded('tfdeploy')
            strategy:
              evaluate: true
              validate: true
            steps:
              - task: TerraformDestroy@0
                displayName: tfdestroy
                inputs:

```

The screenshot shows the Azure Pipeline interface. On the left, there's a preview of a pipeline run with a red arrow pointing to the 'main' branch selection dropdown. In the center, the 'Run pipeline' dialog is open, showing the 'Branch/tag' dropdown set to 'main'. A large red arrow points from the dialog down to the 'Run pipeline' button. Below the dialog, the pipeline's code is visible in a code editor. To the right, the 'Jobs in run #20221228.1' section shows the execution history of the 'apply' job, with each step (validate, apply, plan, apply) showing its status, duration, and logs. On the far right, the Microsoft Azure portal shows the 'myResourceGroup-777' resource group details.

File view with a record of all were created entities

This screenshot shows the Azure Storage Container 'tfstate' containing the 'terrafrom.tfstate' file. The file is a JSON object representing Terraform state. A red arrow points to the file content area, which displays the following code:

```

{
  "version": 4,
  "terraform_version": "1.3.7",
  "serial": 8,
  "lineage": "369b49d1-3ee0-9dd1-6457-c805118da4f5",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "azurerm_app_service",
      "name": "myapp",
      "provider": "provider[\"registry.terraform.io/hashicorp/azurerm\"]",
      "instances": [
        {
          "schema_version": 0,
          "additional_provider_params": {},
          "allow_anonymous_redirect_urls": [],
          "default_provider": "",
          "enabled": false,
          "facebook": [],
          "google": [],
          "microsoft": [],
          "runtime_version": "",
          "token_refresh_extension_hours": 0,
          "token_store_enabled": false,
          "twitter": [],
          "unauthenticated_client_action": ""
        }
      ],
      "auth_settings": [
        {
          "active_directory": [],
          "additional_provider_params": {},
          "allow_anonymous_redirect_urls": [],
          "default_provider": "",
          "enabled": false,
          "facebook": [],
          "google": [],
          "microsoft": [],
          "runtime_version": "",
          "token_refresh_extension_hours": 0,
          "token_store_enabled": false,
          "twitter": [],
          "unauthenticated_client_action": ""
        }
      ],
      "backup": [],
      "client_affinity_enabled": false,
      "client_cert_enabled": false,
      "client_cert_mode": "Required",
      "connection_string": [],
      "custom_domain_verification_id": "3B2445FCCD0E884E2E0FAD62381F100F6A102E26B4FB0996AF9A5BE0E856E",
      "default_site_hostname": "appserviceid-876555.azurewebsites.net",
      "enabled": true,
      "https_only": false,
      "id": "/subscriptions/b124b468-39ee-416c-a1ef-9738f476fa3c/resourceGroups/myResourceGroup-777/providers/Microsoft.Web/sites/myapp",
      "identity": []
    }
  ],
  "kv_vault_reference_identity_id": "SystemAssigned"
}

```

Destroy release of infrastructure

1. Pipeline overview: A red arrow points to an artifact named '_artifact_infrastructure' with the caption 'the same artifact of infrastructure which you want to destroy'.

2. Task configuration: 'Install Terraform latest' task selected.

3. Task configuration: 'Terraform : init' task selected.

4. Task configuration: 'Terraform : plan' task selected.

5. Task configuration: 'Terraform : destroy' task selected.

result in state file

Home > csb100320024a9a95bc | Containers > tfstate >

tfstate Container

Search Upload Change access level ...

Overview Diagnose and solve problems Access Control (IAM) Settings Shared access tokens Access policy Properties Metadata

Authentication method: Access key (Switch to Azure AD User Account)
Location: tfstate

Search blobs by prefix (case-insensitive): Show deleted blobs Add filter

Name

terraform.tfstate

terraform.tfstate ...

Blob Save Discard Download Refresh Delete

Overview Versions Snapshots Edit Generate SAS

The file 'terraform.tfstate' may not render correctly as it contains an unrecognized extension.

```

1 {
2   "version": 4,
3   "terraform_version": "1.3.7",
4   "serial": 9,
5   "lineage": "369b49d1-3ee0-9dd1-6457-c805118da4f5",
6   "outputs": {},
7   "resources": [],
8   "check_results": null
9 }
10

```

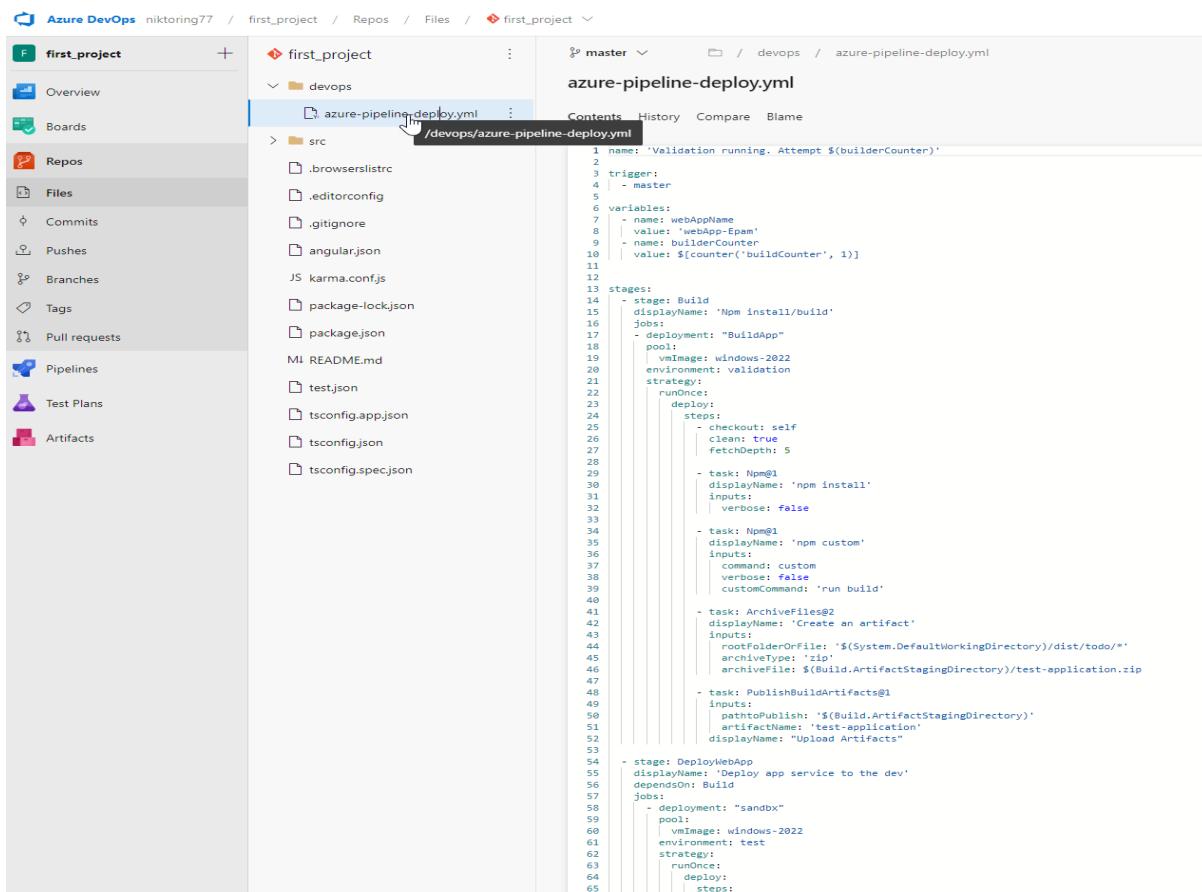
Part 3 – Create a deploy pipeline to app service

1. Add yml pipeline to the application folder
2. Your pipeline structure should contain 2 stages. 1st – build, create zip archive, and publish an artifact. 2nd – download an artifact and deploy it to azure app service
3. To deploy .zip to app service use azure app service deployment task

Service connection – manual way

<https://4bes.nl/2019/07/11/step-by-step-manually-create-an-azure-devops-service-connection-to-azure/>

Don't forget to grant access on the subscription level for your enterprise application (service principal)



The screenshot shows the Azure DevOps interface with the repository structure for 'first_project'. The 'azur...-deploy.yml' file is selected, displaying its YAML content. The code defines a pipeline with two stages: 'Build' and 'DeployWebApp'. The 'Build' stage uses an NPM task to install dependencies and a custom task to run a build command. The 'DeployWebApp' stage uses an ArchiveFiles task to create a zip artifact and a PublishBuildArtifacts task to upload it. Finally, an Azure App Service Deployment task is used to deploy the artifact to an app service.

```
name: 'Validation running. Attempt ${builderCounter}'  
trigger:  
| - master  
variables:  
| - name: webAppName  
| | value: 'webApp-Epm'  
| - name: builderCounter  
| | value: ${counter('buildCounter', 1)}  
stages:  
- stage: Build  
displayName: 'Npm install/build'  
jobs:  
- deployment: "BuildApp"  
pool:  
vmImage: windows-2022  
environment: validation  
strategy:  
runOnce:  
deploy:  
steps:  
- checkout: self  
clean: true  
fetchDepth: 5  
- task: Npm@1  
displayName: 'npm install'  
inputs:  
| verbose: false  
- task: Npm@1  
displayName: 'npm custom'  
inputs:  
| cwd: custom  
| verbose: false  
| customCommand: 'run build'  
- task: ArchiveFiles@2  
displayName: 'Create an artifact'  
inputs:  
| rootFolderOrFile: '$(System.DefaultWorkingDirectory)/dist/todo/*'  
| archiveType: 'zip'  
| archiveFile: '$(Build.ArtifactStagingDirectory)/test-application.zip'  
- task: PublishBuildArtifacts@1  
inputs:  
| pathToPublish: '$(Build.ArtifactStagingDirectory)'  
| artifactName: 'test-application'  
| displayName: "Upload Artifacts"  
- stage: DeployWebApp  
displayName: 'Deploy app service to the dev'  
dependsOn: Build  
jobs:  
- deployment: "sandbox"  
pool:  
vmImage: windows-2022  
environment: test  
strategy:  
runOnce:  
deploy:  
steps:
```

Azure DevOps Pipeline Details:

The screenshots show the following pipeline steps:

- BuildApp Environment:**
 - Npm install/build
 - BuildApp (4m 48s)
 - Initialize job (4s)
 - Checkout first_project@master to s (6s)
 - npm install (3m 10s)
 - npm custom (1m 17s)
 - Create an artifact (1s)
 - Upload Artifacts (2s)
 - Post-job: Checkout first_project... (<1s)
 - Finalize Job (<1s)
 - Deploy app service to the dev
 - sandbox
- sandbox Environment:**
 - Npm install/build
 - BuildApp (4m 48s)
 - Initialize job (4s)
 - Deploy app service to the dev
 - sandbox
 - Finalize Job (<1s)
- Validation Step:**
 - Checks and manual validations for Deploy app service to the dev
 - 1 approval needs your review before the dev
 - 1 Job is pending...
 - Approval status: Waiting
 - Environment: test

Creative Tim Dashboard:

The dashboard includes the following sections:

- Sidebar:** Работа, Семья, Учеба, Отдых, Спорт, Еда, Финансы, Гаджеты, Здоровье, Автомобиль, Ремонт.
- Dashboard Overview:**
 - Used Space: 49/50 GB
 - Revenue: \$34,245
 - Fixed Issues: 75
 - Followers: +245
- Key Metrics:**
 - Daily Sales: 55% increase in today sales.
 - Email Subscriptions: Last Campaign Performance
 - Completed Tasks: Last Campaign Performance
- Task Management:**
 - Tasks: BUGS, WEBSITE, SERVER
 - Sign contract for "What are conference organizers afraid of?" (checked)
 - Lines From Great Russian Literature? Or E-mails From My Boss?
 - Flooded: One year later, assessing what was lost and what was found when a ravaging rain swept through metro Detroit
 - Create 4 Invisible User Experiences you Never Knew About (checked)
- Employee Statistics:**

ID	Name	Salary	Country
1	Dakota Rice	\$36,738	Niger
2	Minerva Hooper	\$23,789	Curaçao
3	Sage Rodriguez	\$56,142	Netherlands
4	Philip Chaney	\$38,735	Korea, South

Azure CLI deploy required Azure resources

```
call az group create --location westeurop --name cloud-shell-storage-westeurope

call az storage account create --name csb100320024a9a95bc --resource-group
cloud-shell-storage-westeurope --location westeurop --sku Standard_LRS

call az storage container create --name terraform --account-name csb100320024a9a95bc

call az storage account keys list -g cloud-shell-storage-westeurope -n csb100320024a9a95bc
```

example 2

```
call az group create --location westeurop --name $(terraformstoragerg)

call az storage account create --name $(terraformstorageaccount) --resource-group
$(terraformstoragerg) --location westeurop --sku Standard_LRS

call az storage container create --name terraform --account-name $(terraformstorageaccount)

call az storage account keys list -g $(terraformstoragerg) -n $(terraformstorageaccount)
```

Azure PowerShell script: InlineScript

```
# You can write your azure powershell scripts inline here.
# You can also pass predefined and custom variables to this script using arguments
```

```
$key=(Get-AzureRmStorageAccountKey -ResourceGroupName $(cloud-shell-storage-westeurope)
-AccountName $(csb100320024a9a95bc)).Value[0]
Write-Host "##vso[task.setvariable variable=storagekey]$key"
```