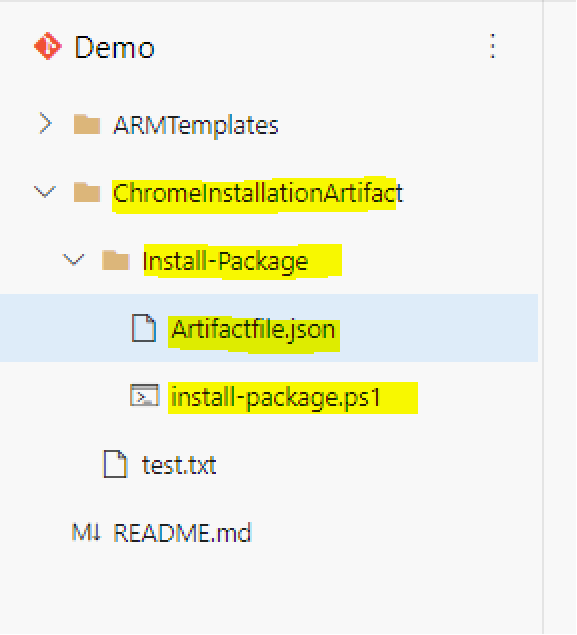
How to add an custom artifact for Dev Test Lab and deploy it with ARM template from pipeline

Step 1: Create an custom Artifact

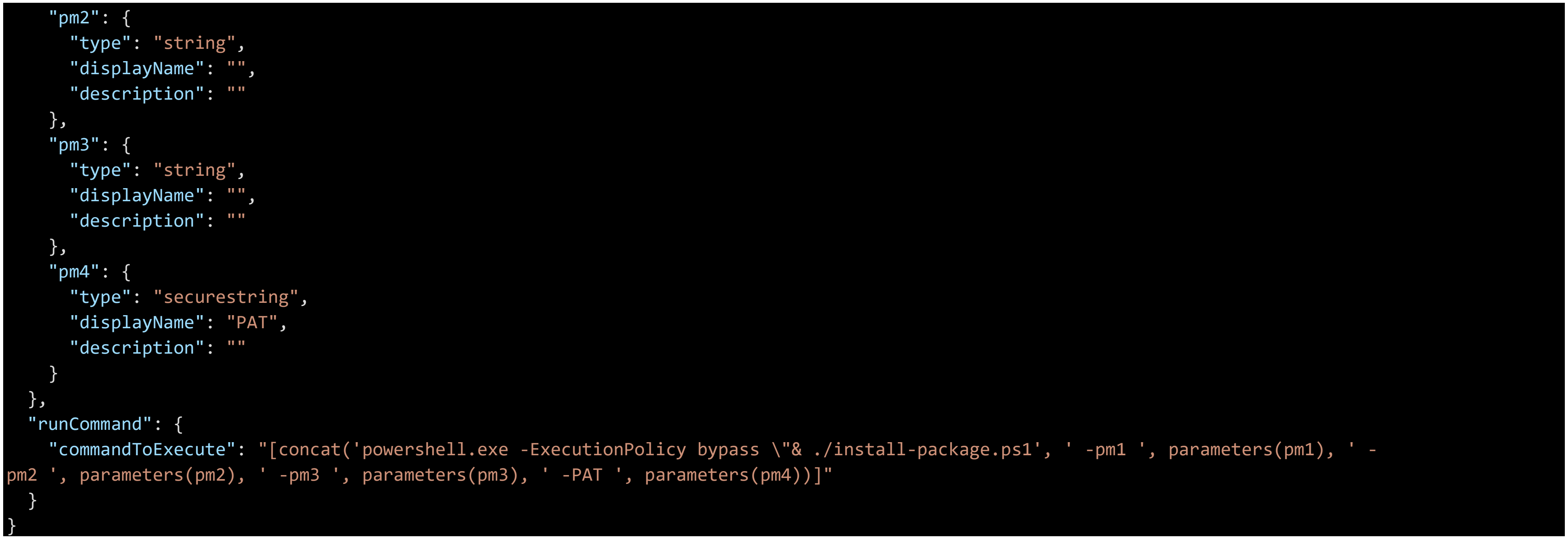
- While creating an custom artifact create a folder with any name.
- Put the .ps1 script and an file "Artifactfile.json" as below. (reference link)



Demo
ARMTemplates
ChromeInstallationArtifact
Install-Package
Artifactfile.json
install-package.ps1
test.txt
README.md

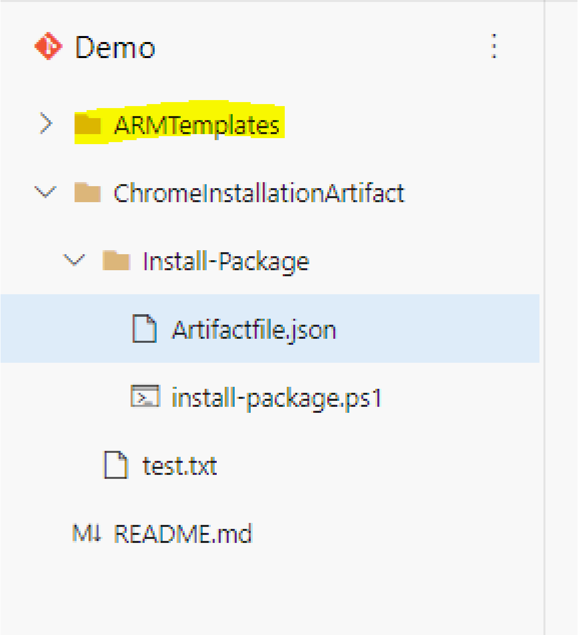
- Configure Artifactfile.json to run the .ps1 file.
  Sample Artifactfile.json will look like:

```
{
  "$schema": "https://raw.githubusercontent.com/Azure/azure-devtestlab/master/schemas/2016-11-28/dtlArtifacts.json",
  "title": "<<Expected title>>",
  "publisher": "Saumya Mishra",
  "description": "Description",
  "tags": [
    "Exchange"
  ],
  "targetOsType": "Windows",
  "parameters": {
    "pm1": {
      "type": "string",
      "displayName": "",
      "description": ""
    },
```



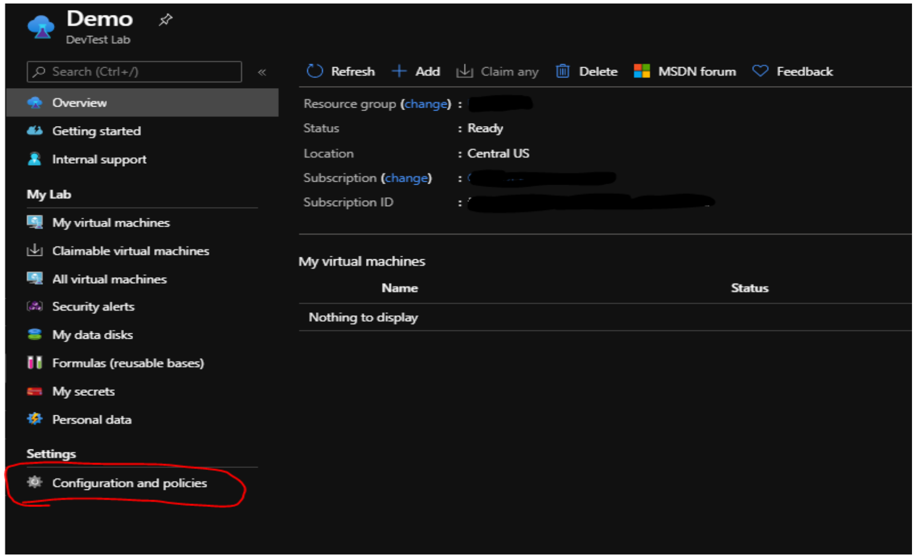
```
    "pm2": {
      "type": "string",
      "displayName": "",
      "description": ""
    },
    "pm3": {
      "type": "string",
      "displayName": "",
      "description": ""
    },
    "pm4": {
      "type": "securestring",
      "displayName": "PAT",
      "description": ""
    }
  },
  "runCommand": {
    "commandToExecute": "[concat('powershell.exe -ExecutionPolicy bypass \"& ./install-package.ps1', ' -pm1 ', parameters(pm1), ' -
pm2 ', parameters(pm2), ' -pm3 ', parameters(pm3), ' -PAT ', parameters(pm4))]"
  }
}
```

- Create an ARM template for which we are going to access the artifact.



Step 2: Add the Artifact to DevTestLab Repository

1) Create a new DevTest Lab
2) Go to configuration and policies:

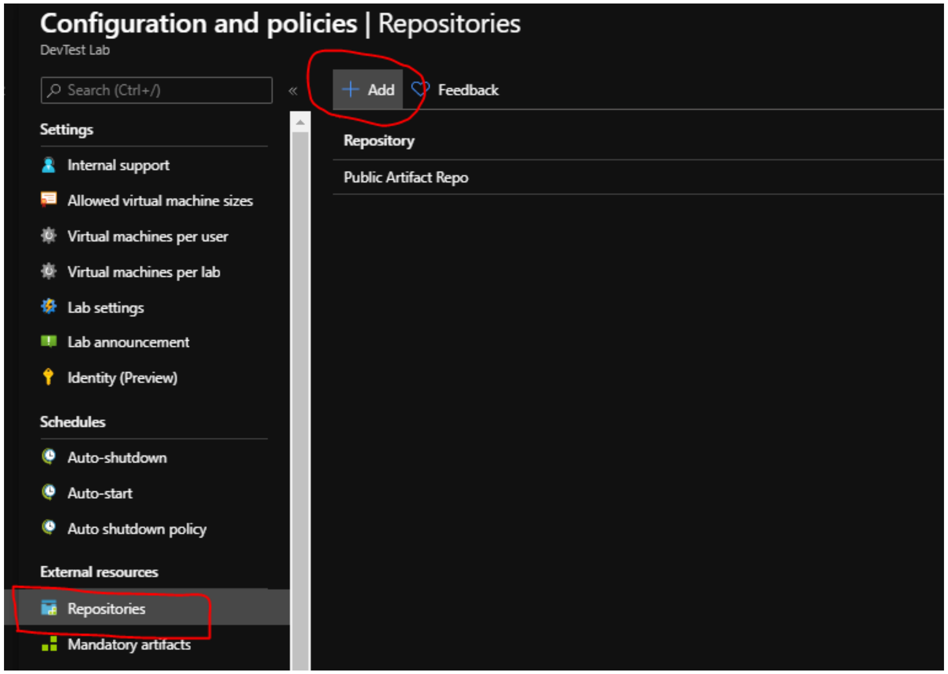


Resource group (change) : [redacted]
Status : Ready
Location : Central US
Subscription (change) : [redacted]
Subscription ID : [redacted]

My virtual machines

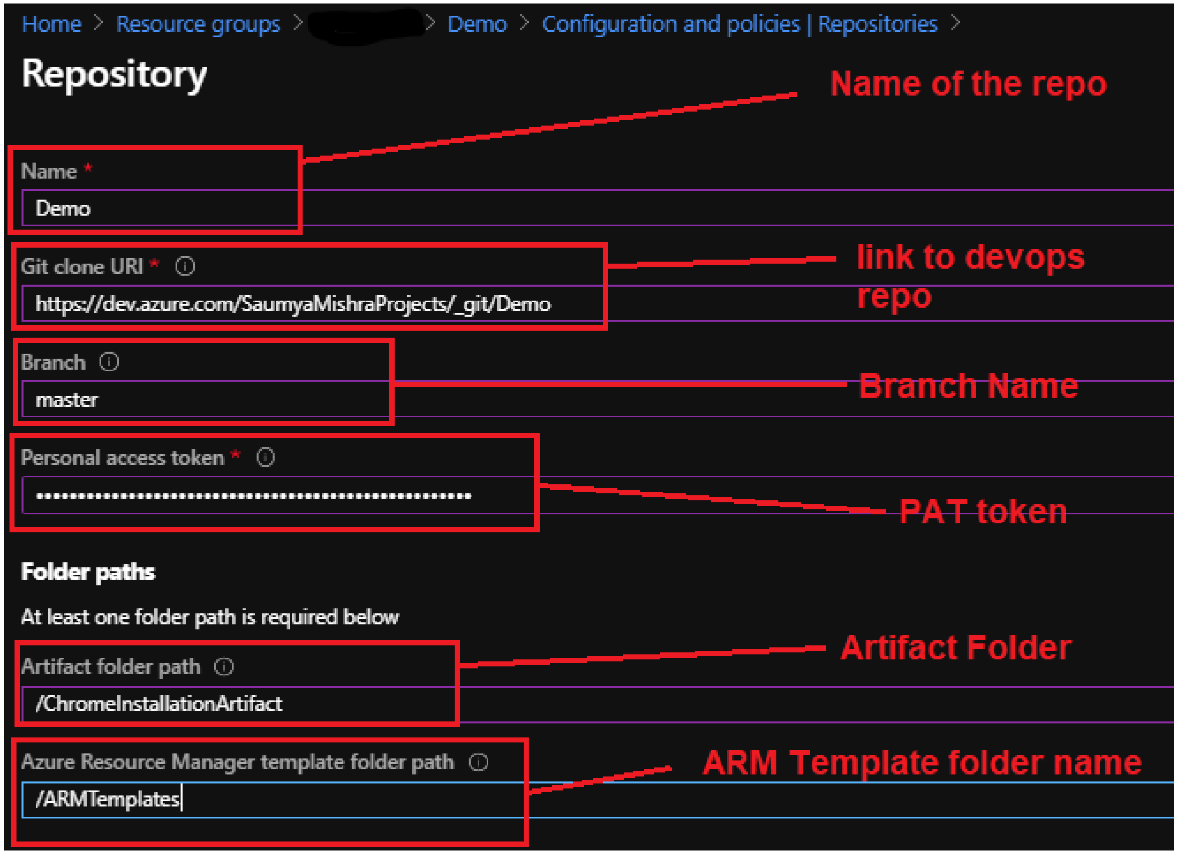
Name	Status
Nothing to display	

3) Add a repository.



Repository
Public Artifact Repo

4) Fill the form to create an artifact



Home > Resource groups > [redacted] > Demo > Configuration and policies | Repositories >

Repository

Name *
Demo

Git clone URI *
https://dev.azure.com/SaumyaMishraProjects/_git/Demo

Branch
master

Personal access token *
••

Folder paths

At least one folder path is required below

Artifact folder path
/ChromeInstallationArtifact

Azure Resource Manager template folder path
/ARMTemplates

Name of the repo

link to devops repo

Branch Name

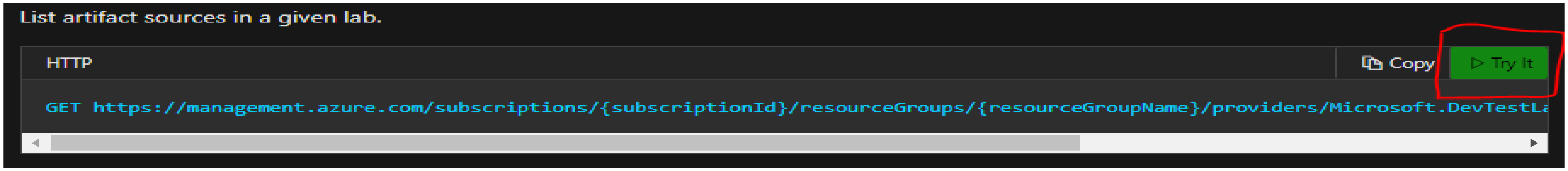
PAT token

Artifact Folder

ARM Template folder name

Step 3 : Get the name (NOT display name) of the artifact.

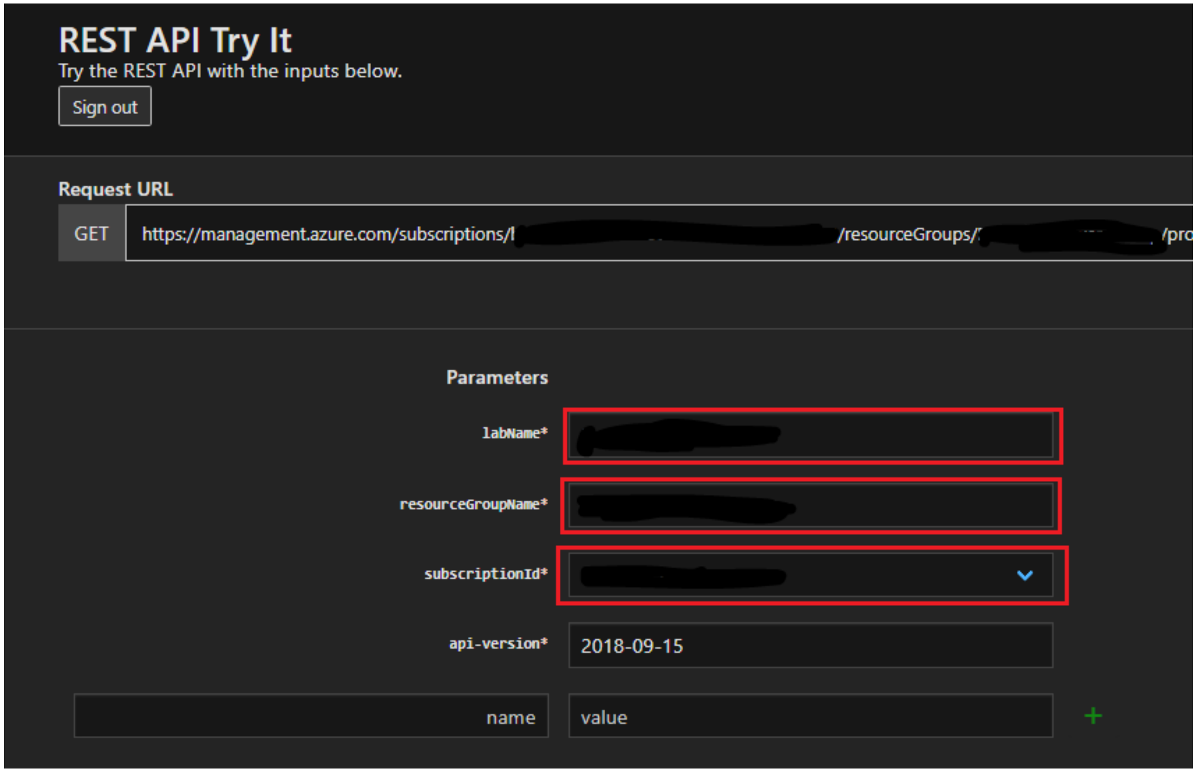
- Goto link and run the command as below, the output of the API will give you the name of the Artifact.



List artifact sources in a given lab.

```
GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.DevTestLa
```

- Login with your account and provide the details as below and you should be able to see the name of the artifact



REST API Try It

Try the REST API with the inputs below.

Request URL

GET https://management.azure.com/subscriptions/[redacted]/resourceGroups/[redacted]/pro

Parameters

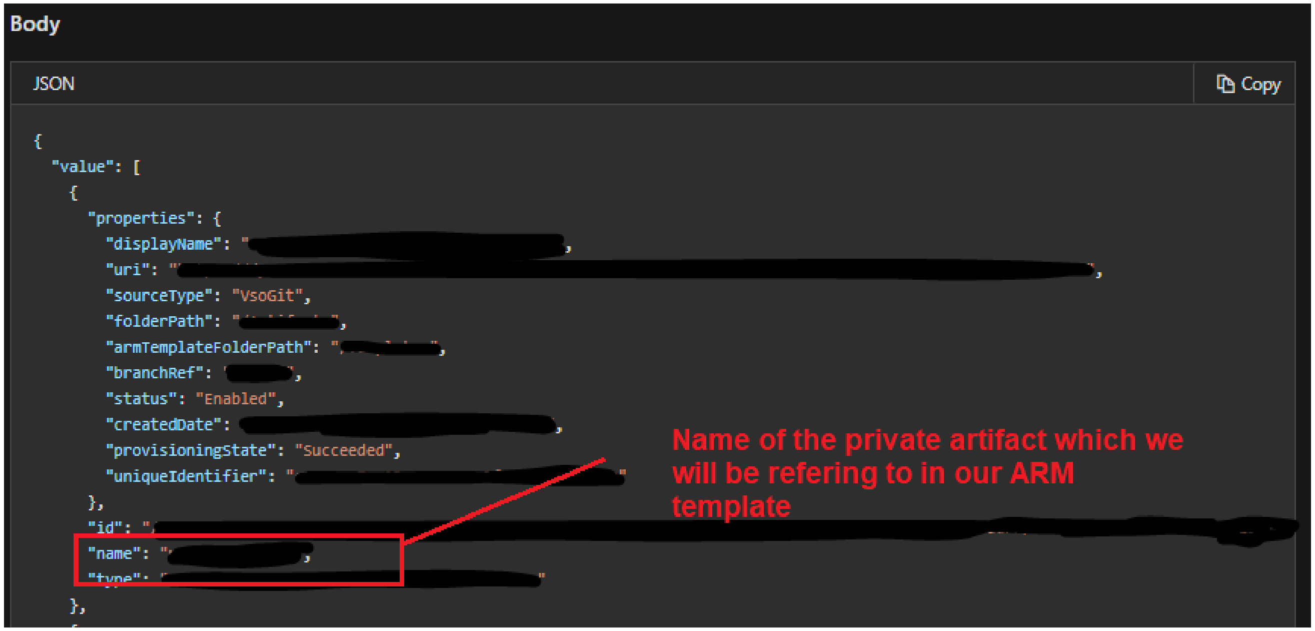
labName* [redacted]

resourceGroupName* [redacted]

subscriptionId* [redacted]

api-version* 2018-09-15

Sample output:



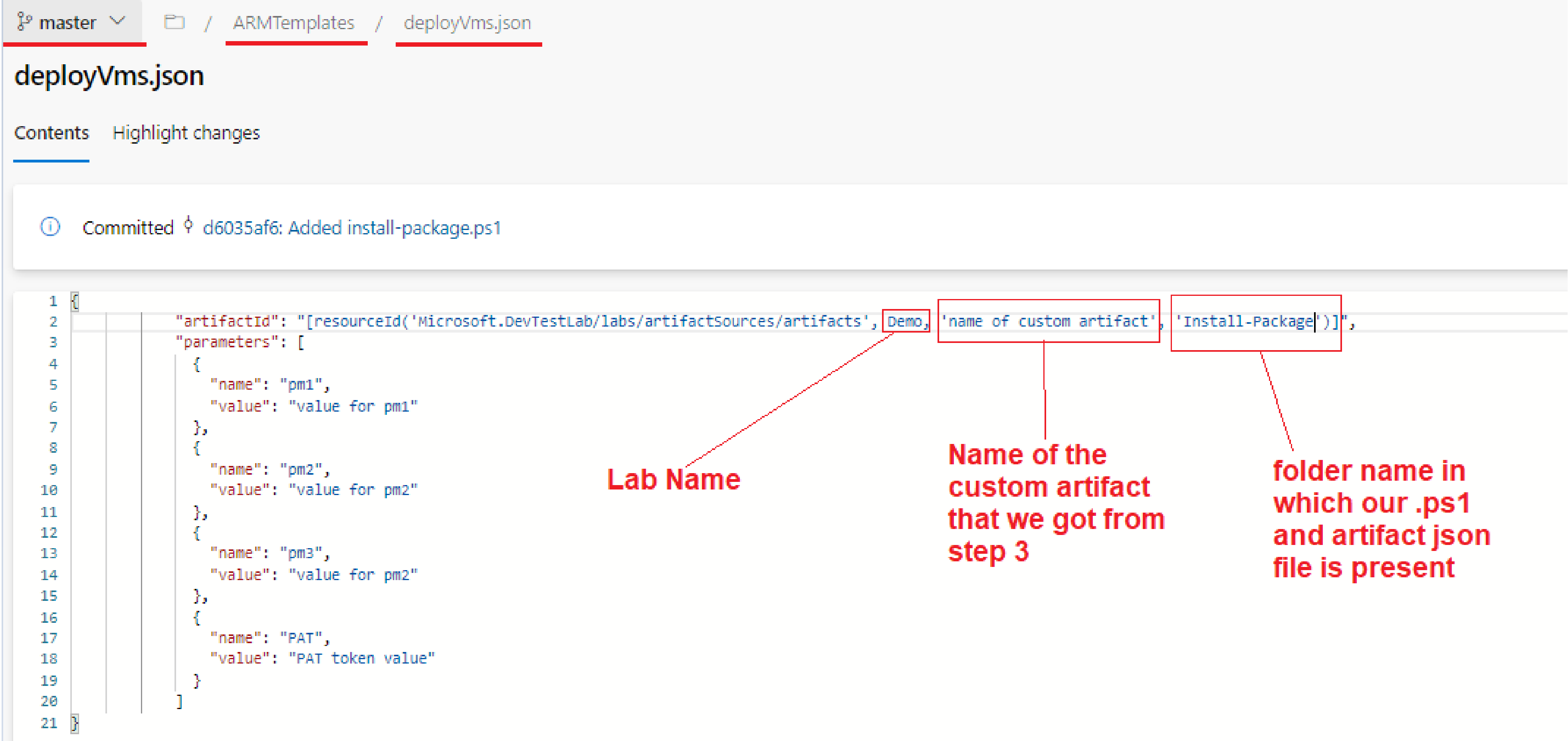
Body

```
{
  "value": [
    {
      "properties": {
        "displayName": "[redacted],
        "uri": "[redacted],
        "sourceType": "VsoGit",
        "folderPath": "[redacted]",
        "armTemplateFolderPath": "[redacted]",
        "branchRef": [redacted],
        "status": "Enabled",
        "createdDate": [redacted],
        "provisioningState": "Succeeded",
        "uniqueIdentifier": "[redacted]"
      },
      "id": "[redacted]
      "name": "[redacted],
      "type": "[redacted]"
    },
```

Name of the private artifact which we will be refering to in our ARM template

Step 4: passing the parameter and referring to the artifact.

- Sample reference to the artifact will look like below:



deployVms.json

Committed d6035af6: Added install-package.ps1

```
{
      "artifactId": "[resourceId('Microsoft.DevTestLab/labs/artifactSources/artifacts', Demo, 'name of custom artifact', 'Install-Package')]",
      "parameters": [
        {
          "name": "pm1",
          "value": "value for pm1"
        },
        {
          "name": "pm2",
          "value": "value for pm2"
        },
        {
          "name": "pm3",
          "value": "value for pm2"
        },
        {
          "name": "PAT",
          "value": "PAT token value"
        }
      ]
}
```

Lab Name

Name of the custom artifact that we got from step 3

folder name in which our .ps1 and artifact json file is present