**Deploy a function app as RESTful API in Azure**

**Steps to deploy function app in Azure:**

1. Follow the steps in <https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-azure-function>.
2. From the Azure portal menu or the Home page, select Create a resource.
3. In the New page, select Compute > Function App.
4. On the Basics page, use the function app settings as specified in the following table.

* Subscription: Microsoft Azure Sponsorship
* Resource group: lninternalconsole
* Function App name: ML\_azure\_api
* Publish: code
* Runtime stack: Python
* Version: 3.7
* Region: West India

1. Select Next: Hosting. On the Hosting page, enter the following settings.

* Storage account:
* Operating system: Windows
* Plan: Consumption (Serverless)/ ASP-lninternalconsole-8712 (PremiumV2)/ SoutheastAsiaLinuxDynamicPlan

1. Select Next: Monitoring. On the Monitoring page, enter the following settings.
2. On the Review + create page, review your settings, and then select Create to provision and deploy the function app.
3. Download & install chocolatey/choco on Windows 10 through powershell (“Run as Administrator“)

Set-ExecutionPolicy Bypass -Scope Process -Force;

iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

1. Deployment of Azure functions from local computer through Azure CLI:

* choco install azure-cli
* choco install python --version 3.6
* choco install azure-functions-core-tools

1. Initiate the Project directory

func init directory-path

1. Create virtual env in required folder where your API resides

py -3.6 -m venv .env

1. Activate virtual environment

.env\scripts\activate

1. Change the directory to project folder

cd directory-path

1. Install Requirements using requirements.txt

pip install -r requirements.txt

1. Create API

func new

choose type of function

write the name of function

1. Paste your function code here and Start the function pasted

func host start

1. Publish to Azure function app

func azure functionapp publish lninternalconsole