



# Azure Spring Cloud on Learn TV

Rapidly dev and deploy Spring apps to  
Azure Spring Cloud

Yev Bronshteyn

---

# Agenda

- 
- Provisioning
  - Building
  - Deploying
  - Best practices

# Three pipelines

A blue rounded rectangle with a slight gradient and a shadow, containing the word "Provision" in white text.

Provision

A purple rounded rectangle with a slight gradient and a shadow, containing the word "Build" in white text.

Build

A green rounded rectangle with a slight gradient and a shadow, containing the word "Deploy" in white text.

Deploy

# Production

Provision

Build



Deploy

Includes rigorous unit tests

Includes rigorous integration tests

# Nightly



# Provision

- Terraform
- Azure Resource Manager

# Terraform

main.tf

```
resource "azurerm_spring_cloud_service" "example" {
  name                = var.spring_cloud_service
  resource_group_name = azurerm_resource_group.example.name
  location            = azurerm_resource_group.example.location

  config_server_git_setting {
    uri          = https://github.com/selvasingh/spring-petclinic-microservices-config
    label        = "master"
    search_paths = ["."]
  }

  tags = {
    Env = "staging"
  }
}

resource "azurerm_spring_cloud_app" "appname" {
  name                = appname
  resource_group_name = azurerm_resource_group.example.name
  service_name        = azurerm_spring_cloud_service.example.name
}
```



Microsoft DevLabs | 15,784 installs | ★★★★★ (6) | Free

Install terraform and run terraform commands to manage resources on Azure, AWS and GCP.

Get it free

Overview

Q & A

Rating & Review

## About Terraform

Terraform is an open-source tool created by HashiCorp for developing, changing and versioning infrastructure safely and efficiently. It provides a service known as "Infrastructure as Code" which enables users to define and provision infrastructure using a high-level configuration language.

## About the Terraform extension

This extension provides the following components:

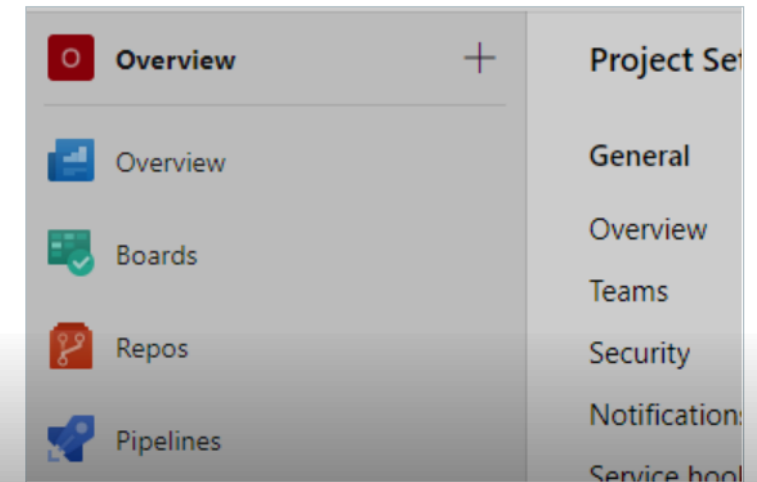
- A service connection for connecting to an Amazon Web Services(AWS) account
- A service connection for connecting to a Google Cloud Platform(GCP) account
- A task for installing a specific version of Terraform, if not already installed, on the agent
- A task for executing the core Terraform commands

The Terraform tool installer task acquires a specified version of [Terraform](#) from the Internet or the tools cache and prepends it to the PATH of the Azure Pipelines Agent (hosted or private). This task can be used to change the version of Terraform used in subsequent tasks. Adding this task before the [Terraform task](#) in a build definition

ensures you are using the right Terraform version.

The Terraform task enables running Terraform commands as part of Azure Build and Release Pipelines providing

- init



## Categories

Azure Pipelines

## Tags

AWS Azure Deploy task DevOps GCP Release Terraform Tool task

## Works with

Azure DevOps Services

# Azure Pipelines Terraform task

aka.ms/terraform-task



# Azure Resource Manager (ARM) Templates

azuredeploy.json

```
{
  "type": "Microsoft.AppPlatform/Spring",
  "apiVersion": "2019-05-01-preview",
  "name": "[parameters('springCloudName')]",
  "location": "westus2",
  "sku": {
    "name": "S0",
    "tier": "Standard"
  },
  "properties": {
    "configServerProperties": {
      "configServer": {
        "gitProperty": {
          "repositories": [],
          "uri": "[parameters('configRepositoryUri')]",
          "searchPaths": []
        }
      }
    },
    "trace": {
      "enabled": false
    }
  }
},
```

```
{
  "type": "Microsoft.AppPlatform/Spring/apps",
  "apiVersion": "2019-05-01-preview",
  "name": "[concat(parameters('springCloudName'), '/api-gateway')]",
  "dependsOn": [
    "[resourceId('Microsoft.AppPlatform/Spring', parameters('springCloudName'))]"
  ],
  "properties": {
    "public": true,
    "activeDeploymentName": "default"
  }
}
```



Provision



Build



Deploy

# Build

1. Run the Maven/Gradle build
2. Copy the artifacts to a staging directory
3. Publish the build artifacts



Provision



Build



Deploy

Deploy



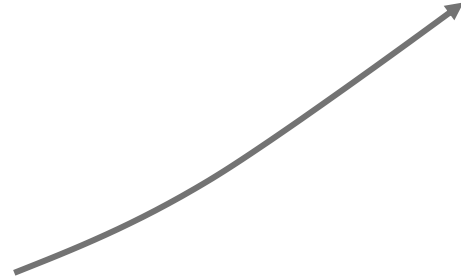
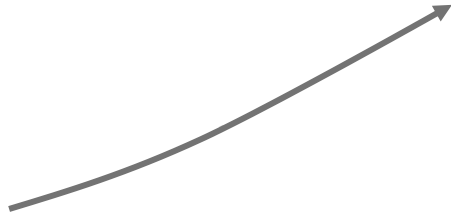
Test



QA



Production



Deploy



d1

d2



Active

Deploy



Active

Deploy





# Azure CLI – Single Deployment

```
bash
```

```
az spring-cloud app deploy \  
    -s "$AZ_SPRING_CLOUD_NAME" \  
    -g "$AZ_RESOURCE_GROUP" \  
    --app "$APP_NAME" \  
    --jar-path $jarPath
```

# Azure CLI – Blue-Green Deployment

```
bash
```

```
az spring-cloud app deployment create \  
  -s "$AZ_SPRING_CLOUD_NAME" \  
  -g "$AZ_RESOURCE_GROUP" \  
  -n "$DEPLOYMENT_NAME" \  
  --app "$APP_NAME" \  
  --jar-path $jarPath
```

```
az spring-cloud app set-deployment \  
  -d "$DEPLOYMENT_NAME" \  
  -g "$AZ_RESOURCE_GROUP" \  
  -s "$AZ_SPRING_CLOUD_NAME"
```

# Best Practices

The background is a dark blue gradient. On the right side, there are several thick, curved lines in shades of blue and cyan. These lines intersect and curve in various directions, creating a dynamic, abstract shape. Small, light blue dots are placed along these lines. In the bottom right corner, there is a faint, repeating pattern of geometric shapes like squares, circles, and lines, creating a textured effect.



Provision



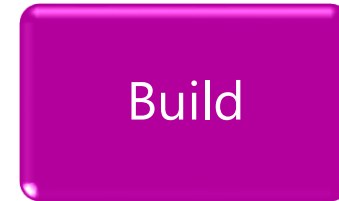
Build



Deploy

# Best Practices

- Loosely-coupled pipelines
  - From environments
  - From each other
- Minimize duplication
  - [Variables](#)
  - [Parameters](#)
  - [Task Groups](#)
  - [Templates](#)
- Multiple pipelines, same YAML



# Best Practices

- Protect Secrets
  - [Secret variable](#)
  - [Secure files](#)
  - [Azure Key Vault](#)
- [Cache the Maven repository](#)

# Sample materials

- [Terraform code](#)
- [Terraform provisioning pipeline](#)
- [ARM template](#)
- [Maven build pipeline](#)

## Get started – build your cloud-native solutions today!

- Get started with Azure Spring Cloud using quickstart: [aka.ms/azure-spring-cloud-start](https://aka.ms/azure-spring-cloud-start)
- Learn using a self-paced workshop on GitHub: [aka.ms/azure-spring-cloud-github](https://aka.ms/azure-spring-cloud-github)
- Learn about implementing solutions on Azure Spring Cloud: [aka.ms/azure-spring-cloud-docs](https://aka.ms/azure-spring-cloud-docs)
- Migrate your apps Azure Spring Cloud
  - Spring Boot: [aka.ms/azure-spring-cloud-migrate-springboot](https://aka.ms/azure-spring-cloud-migrate-springboot)
  - Spring Cloud: [aka.ms/azure-spring-cloud-migrate-springcloud](https://aka.ms/azure-spring-cloud-migrate-springcloud)
  - Tomcat: [aka.ms/azure-spring-cloud-migrate-tomcat](https://aka.ms/azure-spring-cloud-migrate-tomcat)
- Wire Spring apps to interact with Azure services: [aka.ms/spring-integrations](https://aka.ms/spring-integrations)
- For feedback and questions, please reach out to [spring-team@microsoft.com](mailto:spring-team@microsoft.com)





LEARN

[Microsoft.com/Learn](https://Microsoft.com/Learn)