Continuous Deployment with Visual Studio Team Services

Lab Tasks

Task 1: Complete the lab task Continuous Integration with Visual Studio Team Services

1. complete Continuous Integration with VSTS lab.

Task 2: Create a Service Endpoint from VSTS to an Azure Account

1. Create a SP.
    - Install the Azure CLI 2.0, if it is not already installed by following the steps here.
    - Once Azure CLI 2.0 is installed, open Azure CLI 2.0.
    - Log into Azure by running the below command and following the prompts

```
az login
```

    - Create a SP by running the next command.

```
az ad sp create-for-rbac --name DevOpsApp --password Pa$$w0rd01
```

        - Save fields for further use.
    - Check the permissions on the newly created SP and ensure it has a RoleDefinitionName= Contributor, by running the below command.

```
az role assignment list --assignee [app id from earlier] (i.e. 5c0e5186-737c-42ad-881e-16d735ac3dab)
```

    - Log in with the SP to verify access is working fine by running the next command.

```
az login --service-principal -u [app id] --password [password] --tenant [tenant id]
```

        - Where, [app id], [password] and [tenant id] were retrieve when executing command in step 4.
2. Create an Azure Service Endpoint in VSTS.
    - Go to VSTS portal.
    - Click on the Settings icon () at the top menu of the page.
    - Click on Services submenu.
    - Click on <New service connection>.
    - Select <Azure Resource Manager> from the list.
    - In the Add Azure Resource Manager service connection wizard.
        - Click on the link <use the full version of the service connection dialog>.
        - Click the link use the full version of the service connection dialog, that is at the end of the pop up.
        - Fill in the fields required as per the information you obtained earlier when creating your SP. Where:
            - Connection name: DevOpsServiceConnection.
            - Service principal client ID = [app id].
            - Service principal key = [password].
            - Tenant ID = [tenant id].
            - Leave the rest of the fields with their default value or the value filled automatically when adding the Service principal client ID.
        - Click on <Verify connection> link.
        - Ensure you get the green Verified message.
        - Click on <OK>.
    - You should now see the new Endpoint listed.

Task 3: Create a Release Definition

Step 1. Create Azure resources

1. In the azure portal create a Web Application.
    - Go to Azure portal.
    - Click on <Create a Resource>.
    - Select Web -> Web App.
    - Complete:
        - Add the App Name.
        - Select a Subscription.
        - In Resource Group, select <Create new>.
            - Add the new Resource Group name.
        - Change App Service plan.
        - Click on <App Service plan/Location>.
        - Click on <Create new>.
        - Add the App service plan name.
        - Select the closest Location.
        - Click on Pricing tier.
            - Select Production -> S1 tier.
            - Click on <Apply>.
        - In the New App Service Plan, click on <OK>.
    - Leave the rest of the fields with their default value.
    - Click on <Create>.
- When the Web Application is created, from notifications click on <Go to resource>.
- Close the Notifications.
- In the Overview, click on the URL to open the app created.
- Close tab.

Step 2. Add slots on the web App

1. On the App Service click on Deployment -> Deployment Slots from the left menu.
    - Click on <+ Add Slot>.
    - Name: Dev.
    - Click on <OK>.
    - Click on the new Slot.
    - In the Overview (make sure the blade is pointing to the development environment), click on the URL.
    - See the content of the development environment.
    - Close tab.
2. Add a second slot with the Staging name.
    - Click on <+ Add Slot>.
    - Name: Staging.
    - Click on <OK>.
    - Click on the new Slot.
    - In the Overview (make sure the blade is pointing to the staging environment), click on the URL.
    - See the content of the development environment.
    - Close tab.

Step 3. Publish application in Azure

1. Get publish settings.
    - In the azure portal, open the web app created in step 1.
    - Go to Deployment -> Deployment Slots.
    - Click on the <dev> environment.
    - Click on <Get publish profile> from the top menu of the Overview (make sure you are in the dev slot).
    - Click on <Save as> and save the publish Settings file to your computer.
2. Publish VS solution.
    - Open the solution created on Task 1 (CI lab).
    - Right click on the project from the Object Explorer.
    - Click on <Publish…>.
    - From the Publish window, select <Import profile> (if not visible, click on the right arrow).
    - Click on <Publish>.
    - Search and select the publish Settings file you saved from the Web Application.
    - Click on <Open>.
    - A new page will be opened.
    - Make sure the URL is the same of the development environment from the web Application.
3. Go back to the azure portal.
    - Click on the URL (make sure the development environment is selected)
    - Check the version from your VS solutions is published.
    - Close tab.

Step 4. Create a Release Definition to Deploy Infrastructure and Deploy to Dev

1. In VSTS, click on Pipelines -> Releases menu.
2. Click on <+ New pipeline> to create a new definition. This will launch a wizard prompting you to select a deployment template.
    - Click on <Empty job> to start with an empty release.
    - Click on <Apply>.
3. On Stage.
    - Change Stage name: Development.
    - Close the pane on the right-hand side.
4. Configure the source package.
    - Under the Pipeline tab in the Artifacts section, click on <+ Add an artifact>.
    - Enter the below settings in the <Add artifact> pane.
        - Source Type = Build.
        - Project = select your VSTS project name from the list.
        - Source (build pipeline) = select the name of your build definition from the list.
        - Default version = Latest.
        - Source alias = the build source.
    - Click on <Add>.
5. Add a task.
    - Under the Pipeline tab in the Stages section, click on <1 job, 0 task> link.
    - Click on <+> next to the Agent job from the left panel to add a task for this environment.
    - Under Deploy tab, select <Azure App Service Deploy>

- Click <Add>.

6. Configure task.
    - Click on the <Azure App Service Deploy> task from the left panel.
    - Configure it as follows:
        - Version = <default>.
        - Display name = <default>.
        - Azure Subscription: Select the Azure subscription endpoint that you created in task 2.
        - App type = Web App.
        - App Service name = Select the App Service you created in the portal in task 3, step 1.
        - Deploy to slot: Checked.
        - Resource group = Resource group name you created in the portal in task 3, step 1.
        - Slot = Dev (This will deploy the site to the "dev" deployment slot. This allows you to deploy the site to an Azure deployment slot without affecting the Production site).
        - Virtual application = <default>.
        - Package or folder = <default>.
        - In the section Additional Deployment Options.
            - Take App Offline = Checked (This stops the website for deployment period and takes it back online afterwards. This should ensure we get no files locked or in use that we can't overwrite when we deploy to an existing site. It is required because sites receive requests all the time causing files to lock down (i.e. making them unmodifiable)).
            - Publish using Web deploy = <default> (If you were to receive file locked errors, you could check this checkbox and specify a value here such as -retryInterval:6000 -retryAttempts:10, however you do not need to enter this value to deploy the app and can leave it blank).
        - In the section Control Options.
            - Enabled = Checked.
    - Click on <Save> from the top right menu.
    - Add a comment.
    - Click on <OK>.

Step 5. Test the Deployment

1. Click on <+ Release> from the top right menu.
2. Click on <Create a release> to start a new release.
3. From the new panel.
    - Select the environment to trigger (Dev).
    - In Artifacts, select the latest build from the list.
    - Add a Release description.
    - Click on <Create> to start the release.
4. Click the Release-x link from the top message in yellow, to open the release.
5. Hover on the Development Stage.
    - Click on <Deploy> to start deployment manually.
    - Add a comment.
    - Click on <Deploy>. The Deployment will be sent to a queue and then start.

6. Hover on the Development Stage.
    - Click on <Log> to monitor the deployment status.
7. You should see a successful release after a few minutes.
8. If you log into the Azure Portal, you will see the new version of the app application in the development slot.

Step 6. Clone the Dev environment to Staging and Production

1. Clone Development environment to Staging and Production.
2. Open the Release definition to edit it.
   Note: It is possible to change the definition for a Release without changing the Release Definition (i.e. the Release is an instance of the Release Definition that you can edit). You want to make sure that you are editing the Release Definition, not a Release.
    - In VSTS, click on Pipelines -> Releases menu.
    - Click on the new release name from the left panel.
    - Click on <Edit> which is beside the release name on the top of the page.
3. Clone Development environment to Staging.
    - In the release definition click on the Pipeline tab.
    - Hover the mouse over the Development environment.
    - Click Clone icon underneath the Development box (this is hidden until you hover over the environment box).
    - A new Environment is created, which is the copy of the development environment.
    - Click on it to configure Staging environment.
        - Stage name: Staging.
        - Click the X in the right-hand corner of the Environment dialogue to close it.
    - Select the Pre-deployment conditions (the lightning bolt and user icon on the left side of the Staging environment).
        - Select trigger: After stage.
        - Stages: Development.
        - Pre-deployment approvals: Enabled.
        - Type the name of the approver you would like (it can be just you or more) and select them.
            Pre-deployment approvers must approve a deployment coming into the environment. The deployment will stop and wait before executing any tasks in the environment until approval is granted.
            Approvers can be individuals or groups.
        - Click the X to close the Pre-deployment conditions pane.
    - Select the Tasks tab.
    - Choose Staging from the list of Tasks (or on the staging environment box, click the link <1 job, 1 task> to go to the task list.
    - Select <Azure App Service Deploy: [DevOppApp]>.
        - Change Slot to staging.
    - Return to the Pipeline tab.
    - Configure approvers for post-deployment in the Staging environment.
        Earlier we paused the deployment coming in, so we configured pre-deployment approvers, to ensure that if someone is testing in the Staging environment, they don't suddenly get a new build unexpectedly.
        - Click on Post-deployment condition icon (the user icon on the right side of the Staging environment).

    - Post-deployment approvals: Enabled.
    - Type the name of the approver you would like (it can be just you or more) and select them.

      Post-deployment approvers approve deployments so that the next Environment can begin. They act as sign-off for the current environment. Approvers can be individuals or groups.

4. Clone Staging environment to Production.
    - In the release definition click on the Pipeline tab.
    - Hover the mouse over the Staging environment.
    - Click Clone icon underneath the Staging box (this is hidden until you hover over the environment box).
    - A new Environment is created, which is the copy of the staging environment.
    - Click on it to configure Staging environment.
        - Stage name: Production.
        - Click the X in the right-hand corner of the Environment dialogue to close it.
    - Select the Pre-deployment conditions (the lightning bolt and user icon on the left side of the Production environment).
        - Select trigger: After stage.
        - Stages: Staging.
        - Pre-deployment approvals: Enabled.

          Update approvers if necessary.
    - Click the X to close the Pre-deployment conditions pane.
    - Select the Tasks tab.
    - Choose Staging from the list of Tasks (or on the production environment box, click the link <1 job, 1 task> to go to the task list.
    - Select <Azure App Service Deploy: [DevOppApp]>.
        - Deploy to Slot: Uncheck (the site will be deployed to the production slot).
    - Return to the Pipeline tab.
    - Configure approvers for post-deployment in the Production environment.
        - Click on Post-deployment condition icon (the user icon on the right side of the Staging environment).
        - Post-deployment approvals: Enabled.
    - Update approvers if necessary.

Step 7. Configure Continuous Deployment for this Release Definition

1. In the Artifacts section. Click on the lightning icon to enable Continuous Deployment trigger.
2. In the new panel, enable Continuous deployment trigger.
3. Add a Build branch filter.
4. Click on <+ Add>.
    - Type: Include.
    - Build branch: master.
5. Save the Release Definition.
    - Click on <Save> from the top right menu.
    - Add a Comment.
    - Click on <OK>.

Task 4: Create a Release

1. Now that you have configured the Release Pipeline, you are ready to trigger a complete release.
2. Click on <+ Release> to create a new Release.
    - Click on <+ Create a release>.
3. In the new panel:
    - Select the latest build.
    - Click on <Create>.
4. Click the Release-x link from the top message in yellow, to open the release. You will see the progress of the release in the different environments.
5. Once the Development stage has completed the deployment.
    - You will see a notification that an approval is pending.
    - You will also have received an email notification if you ticked "Send an email notification to the approver whom the approval is pending on" option in settings for this environment).
    - Check the development slot of the Azure portal to ensure that the Dev environment is released.
    - Click <Approve> back to the VSTS portal. Optionally enter a comment and click on <Approve>.
    - This will trigger the release into the Staging environment.
6. Once the Staging deployment has completed, you will need to approve that this staging is OK.
7. This will then trigger the pre-approval for Production.
8. Once you have approved the production environment, deployment into the Production environment will begin.
9. To see all your releases and where they are in their respective pipelines, click on <All Releases> and then click the <Overview> link.

Task 5. Trigger a full flow

Now that you have configured the CI and CD process, you are going to test the whole flow.

Step 1. Add a change to your solution

1. Go to VS and open the project created in the Lab CI.
2. Navigate to Solution -> Project -> Views -> Home.
3. Open Index.cshtml file.
4. Change the title and save the change.
5. In the Solution Explorer right click in project.
6. Click on <Commit>.
7. Add a comment.
8. Click on <Commit All>.
9. Click on Sync from the new message or the home page in the Team Explorer.
10. Click on <Push>.

Step 2. Validate build

1. In VSTS go to Pipelines -> Build.
2. Select the last build created.
3. Make sure the new build finishes successfully.

Step 3. Validate release

1. In VSTS go to Pipelines -> Release.
2. Make sure the new release has started.
3. Approve the Staging pre-approval.
4. Go to the development environment URL and validate the new version is published.
5. Approve the Staging post-approval.
6. Approve the Production pre-approval.
7. Go to the staging environment URL and validate the new version is published.
8. Approve the Production post-approval.
9. Go to the production environment URL and validate the new version is published.

References

https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest
https://microsoft.github.io/PartsUnlimited/cicd/200.3x-CICD-M03-CDwithVSTS.html