Intro to the AWS Toolkit for Visual Studio

Self-Paced Lab

Version 1.0

Duration: 45 minutes

Purpose & Background

The purpose of this lab is to install and configure the AWS Toolkit for Visual Studio, and to introduce the capabilities of the AWS Toolkit for Visual Studio.

Lab Exercises

The following exercises should be completed in order for this lab:

1. Install the AWS Toolkit for Visual Studio
2. Credential Setup with the AWS Toolkit for Visual Studio
3. Work with AWS Services through the AWS Toolkit for Visual Studio

Prerequisites

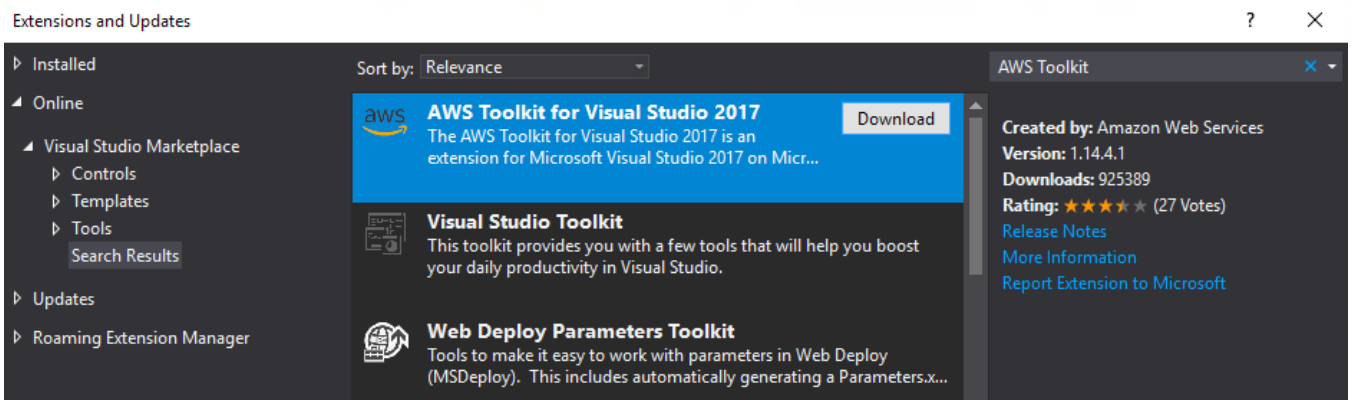
The following are the prerequisites required in order to complete the lab:

- Microsoft Visual Studio 2017 or above installed on your computer
- Internet connection
- AWS Account

Part 1 – Install the AWS Toolkit for Visual Studio

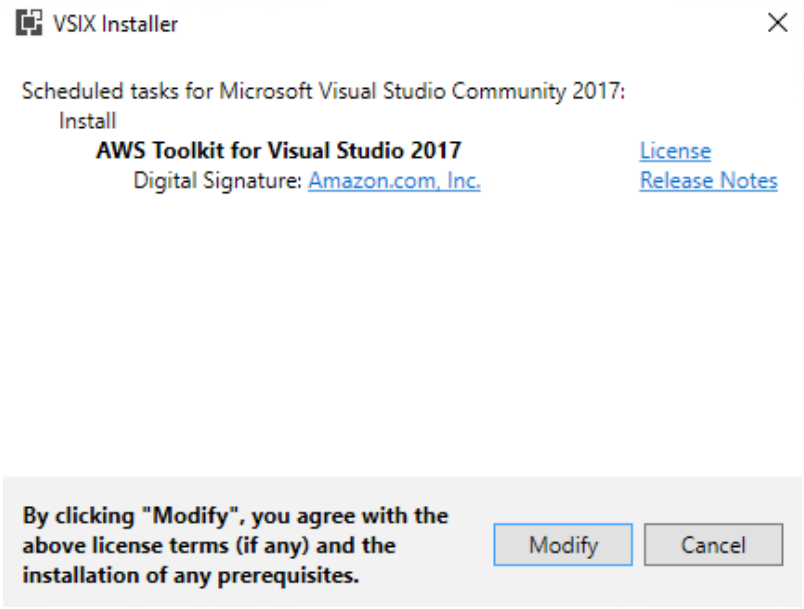
Follow the steps below to install the AWS Toolkit for Visual Studio

1. In Visual Studio, go to Tools -> Extensions and Updates. The Extensions and Updates dialog should open.
2. Select the Online node in the left-panel and then in the search box enter: "AWS Toolkit" (without quotations). Then press ENTER to search.
3. Click the Download button for the AWS Toolkit for Visual Studio 2017.

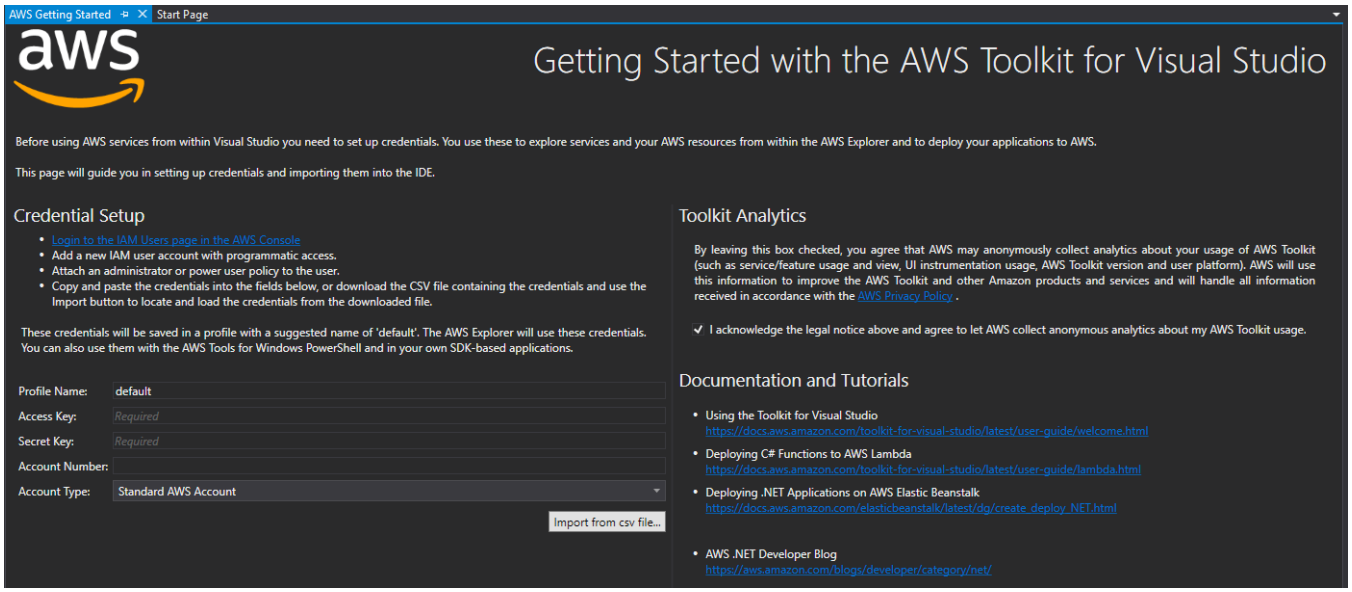


4. Once the AWS Toolkit for Visual Studio has downloaded, a message will appear at the bottom of the Extensions and Updates dialog window letting you know that your changes will be scheduled. Proceed to close the Extensions and Updates dialog window and exit Visual Studio.

5. Upon exiting Visual Studio, a VSIX Installer screen will appear to install the AWS Toolkit. To proceed with the installation, click Modify.



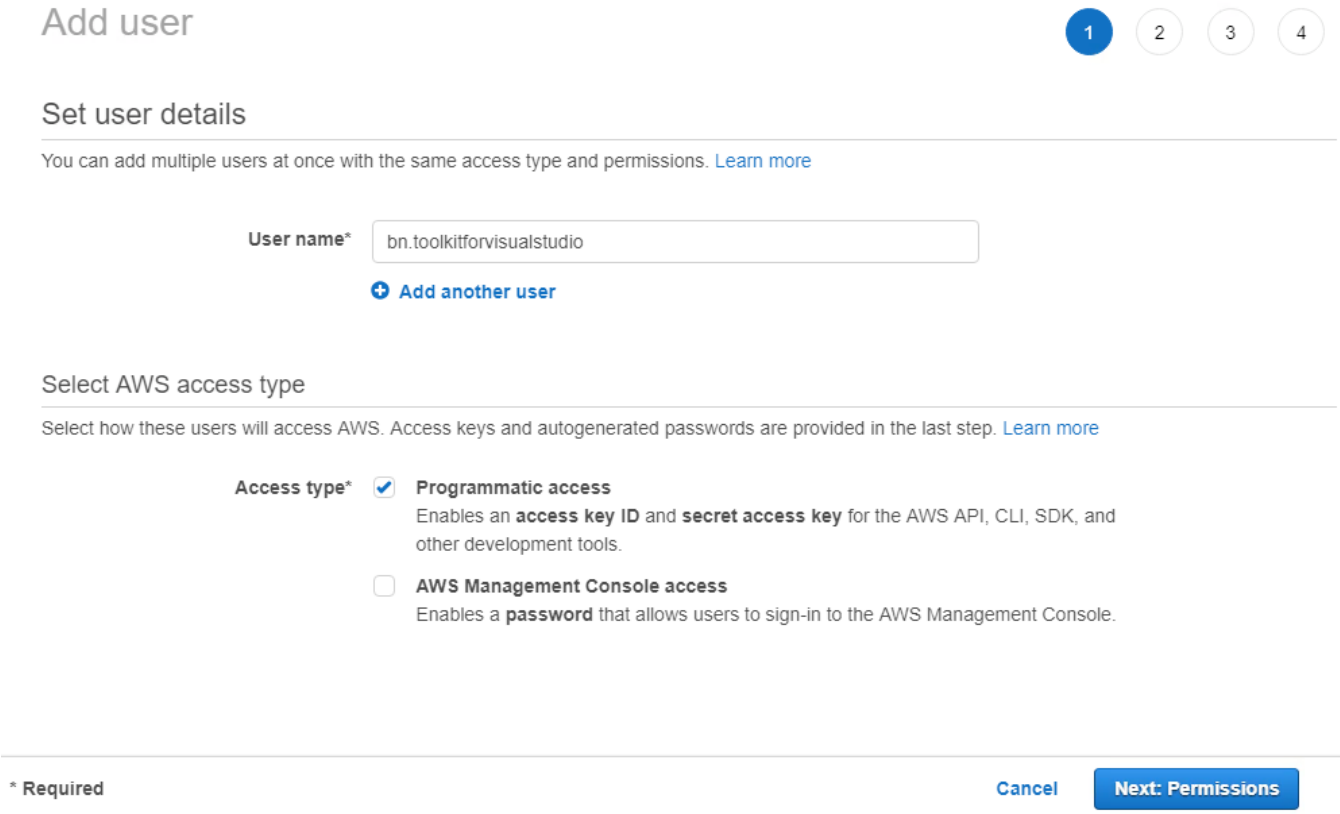
6. Once the installation has successfully completed, close the VSIX Installer and re-launch Visual Studio. You will be greeted by a Getting Started with the AWS Toolkit for Visual Studio screen – this means that you successfully installed it, and are ready to move onto the next part.



Part 2 – Create AWS Credentials for use with the AWS Toolkit for Visual Studio

Follow the steps below to create AWS Credentials for use with the AWS Toolkit for Visual Studio.

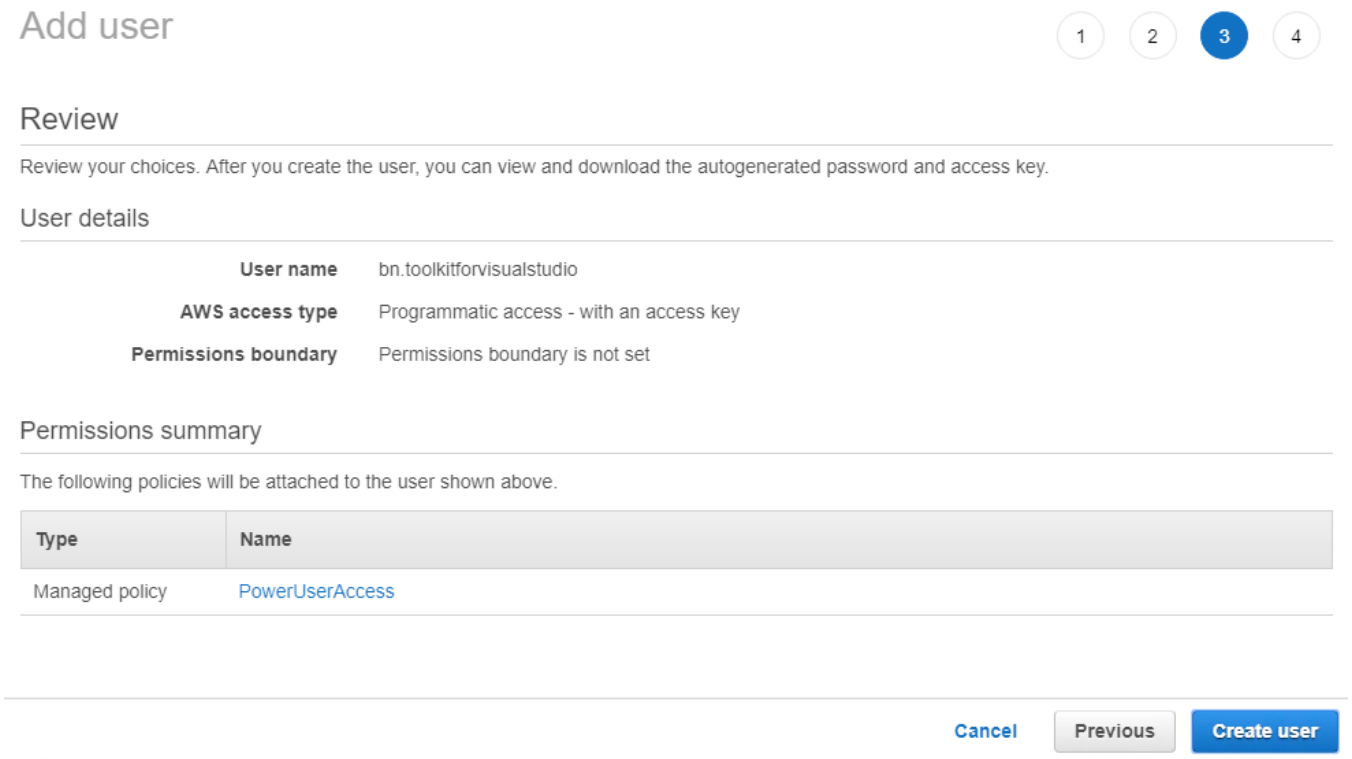
1. Login to the IAM Users page in the AWS Console. You can access this page by visiting https://console.aws.amazon.com/iam/home#/users in your internet browser.
2. Click the Add User button at the top of the page.
3. On the initial Add User screen, set the User Name to your initials dot toolkitforvisualstudio (i.e. bn.toolkitforvisualstudio). For AWS Access Type, check the box for Programmatic Access, and then click the Next: Permissions button.



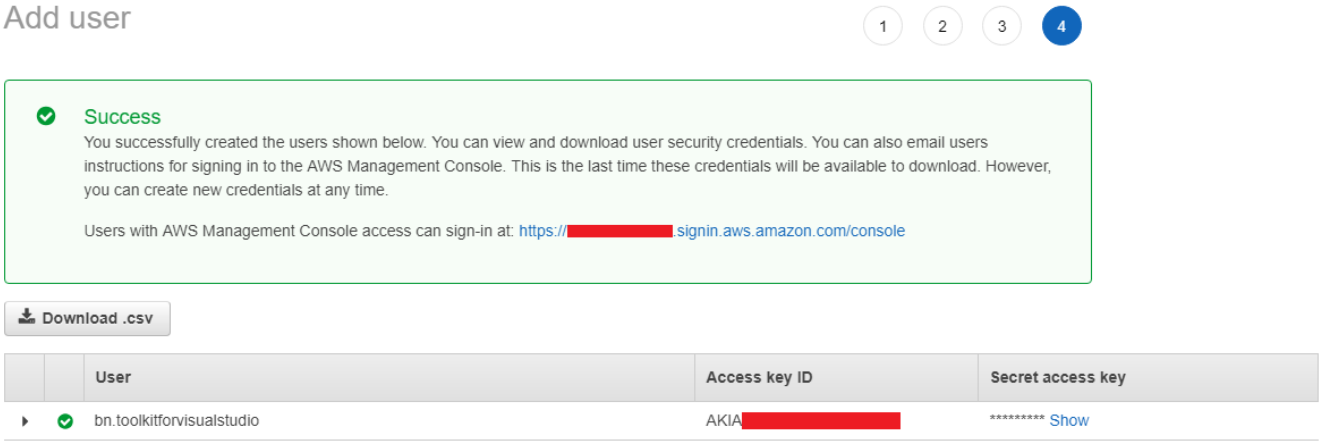
4. On the Set Permissions screen, select Attach Existing Policies Directly, and search for: "PowerUserAccess" (no quotations). Put a checkmark in the Policy Name PowerUserAccess and click the Next: Review button.

5. Review the details on the Review screen to ensure that everything is correct before clicking the Create User button.

    a. Note: If you notice an errors or mistakes, use the Previous button to correct them before creating the user.

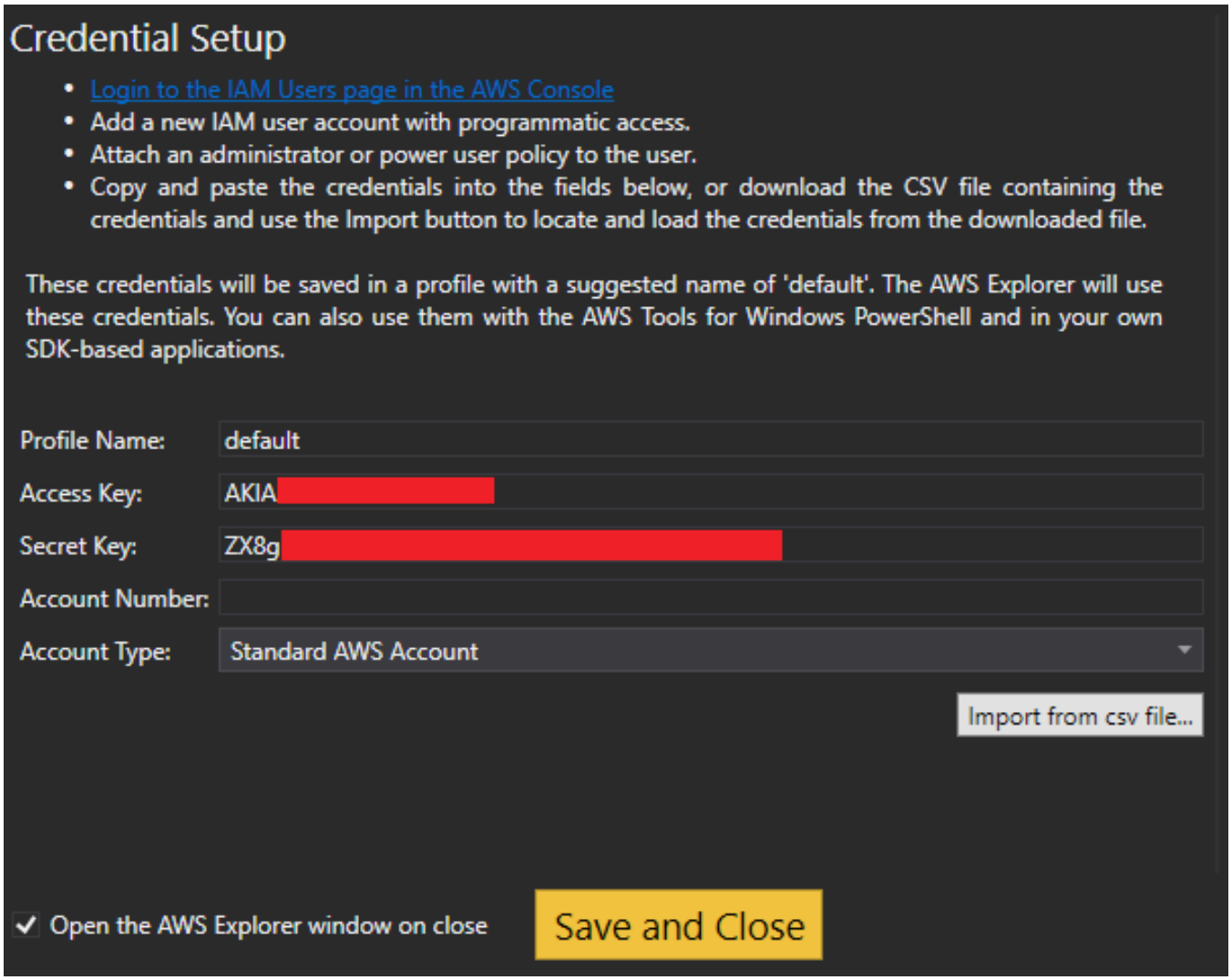


6. On the Success screen, click the Download .csv button to download a CSV file containing your newly created Access Key ID and Secret Access Key.



7. Back in Visual Studio, click the Import from csv File button under the Credential Setup section and select the credentials.csv file you just downloaded from IAM Management Console.

8. You will see that the Access Key and Secret Key auto-fill. Click the Save and Close button at the bottom of the Credential Setup section.



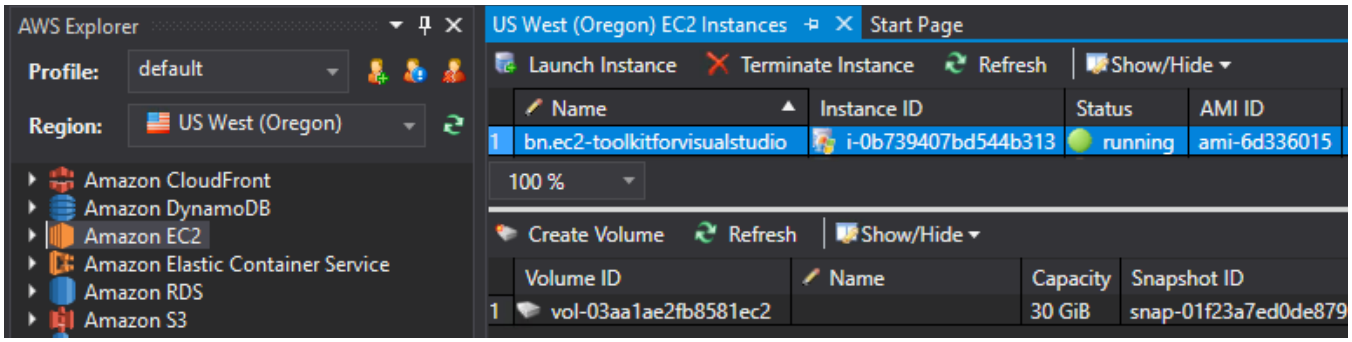
9. Congratulations! You have successfully configured the default Profile for authenticating and accessing AWS through the AWS Toolkit for Visual Studio.

Part 3 – Work with AWS Services through the AWS Toolkit for Visual Studio

For this lab, we'll access and work with AWS Services through the AWS Toolkit for Visual Studio.

1. Now that the AWS Toolkit for Visual Studio has been configured and has valid credentials for your AWS Account, you will be able to interact with AWS resources and services directly within Visual Studio. The AWS Explorer pane in Visual Studio consists of the following elements:
    a. Profile: A list of configured and available AWS Credential Profiles (i.e. default)
    b. Region: A list of available regional AWS service endpoints (i.e. US West (Oregon))
    c. Services: A list of AWS services that you can interact with through the toolkit
2. In the AWS Explorer pane, right-click on Amazon EC2 and select Launch Instance.
3. In the Quick Launch window,
    a. OS: Microsoft Windows Server 2016 Base
    b. Type: t2.medium
    c. Name: Set the name to your initials dot ec2-toolkitforvisualstudio (i.e. bn.ec2-toolkitforvisualstudio)
    d. Key Pair: If you have an available Key Pair in your AWS Account, select it – otherwise, select <No key pair>.
    e. VPC Subnet: Under the VPC your created in Part 1, select one of the Public Subnets created in Part 1.
    f. Security Group: ingress-sg
    g. IAM Role: None
    h. Volume Type: General Purpose (SSD)
    i. Size (GB): 30
4. Once the above has been set, click Launch.
5. In the Amazon EC2 Instances window, you will be able to see that your EC2 instance has launched and is now Running. You can see information about the EC2

instance in the top portion, and beneath you will be able to see details about the EBS Volume attached to it.



6. When you right-click on the newly launched EC2 instance, you are able to perform items such as Get Windows Passwords, Open Remote Desktop, and other instance related tasks. Let's go ahead and Terminate this EC2 instance by selecting

Terminate in the context menu. When prompted to confirm the terminate action, click Yes.

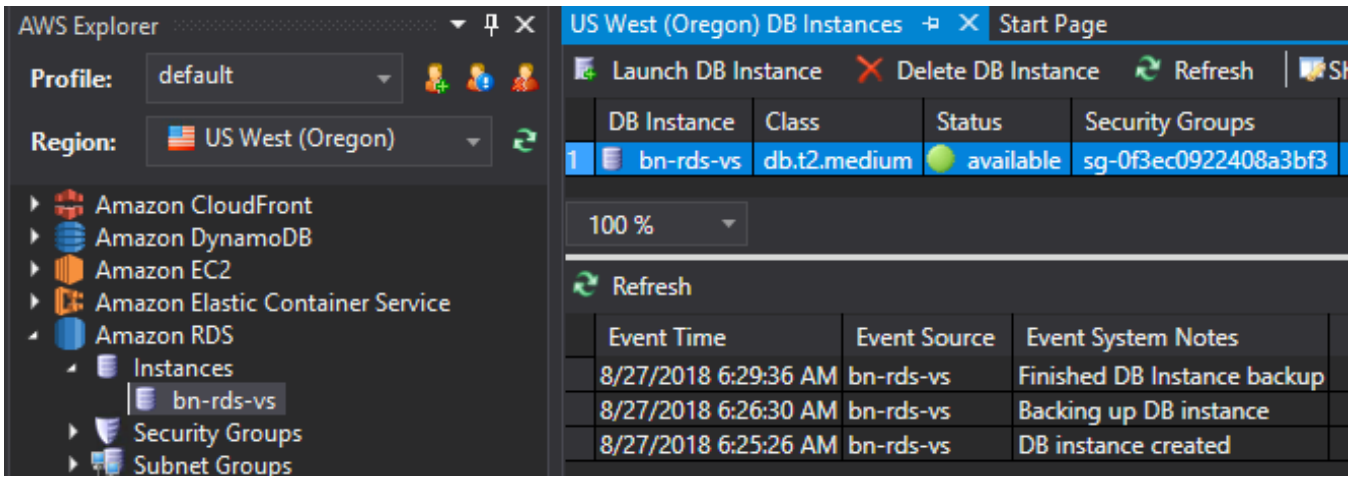
7. In the AWS Explorer pane, right-click on Amazon RDS and select Launch Instance.
8. In the Launch DB Instance window,
    a. Engine Selection
        - Database Engine: Microsoft SQL Server Express Edition
    b. DB Engine Instance Options
        - DB Engine Version: SQL Server 2017 (any version)
        - DB Instance Class: db.t2.medium
        - Perform a multi AZ deployment: Unchecked
        - Upgrade minor versions automatically: Checked
        - Allocated Storage: 20 GB
        - DB Instance Identifier: Set the identifier to your initials dash rds-vs (i.e. bn-rds-vs)
        - Master User Name: master
        - Master User Password: password
        - Confirm Password: password
    c. Advanced Settings
        - VPC: *Select the VPC you created in Part 1*
        - Subnet Group: *Select the Public DB Subnet Group created in Part 1*
        - Publicly Accessible: Checked
        - Availability Zone: No Preference
        - Security Group: *Select the Security Group with Ingress Rule created in Part 1*
        - Database Name: *Leave Blank*
        - Database Port: 1433
        - DB Parameter Group: *Leave Blank*
        - Option Group: *Leave Blank*

d. Backup and Maintenance

- Leave the default values, and just click Next

9. On the Review portion of the Launch DB Instance window, review the details you have inputted and click Launch.

10. In the Amazon RDS Instances window, you will be able to see that your RDS instance has launched and is now Available. You can see information about the RDS instance in the top portion, and beneath you will be able to see Recent Events concerning this RDS instance.



11. When you right-click on the newly launched RDS instance, you are able to perform items such as Add to Server Explorer, Create SQL Server Database, and other instance related tasks. Let's go ahead and Delete this RDS instance by selecting Delete DB Instance in the context menu. When prompted to confirm the delete action, ensure that Create Final Snapshot is unchecked, and click OK.