

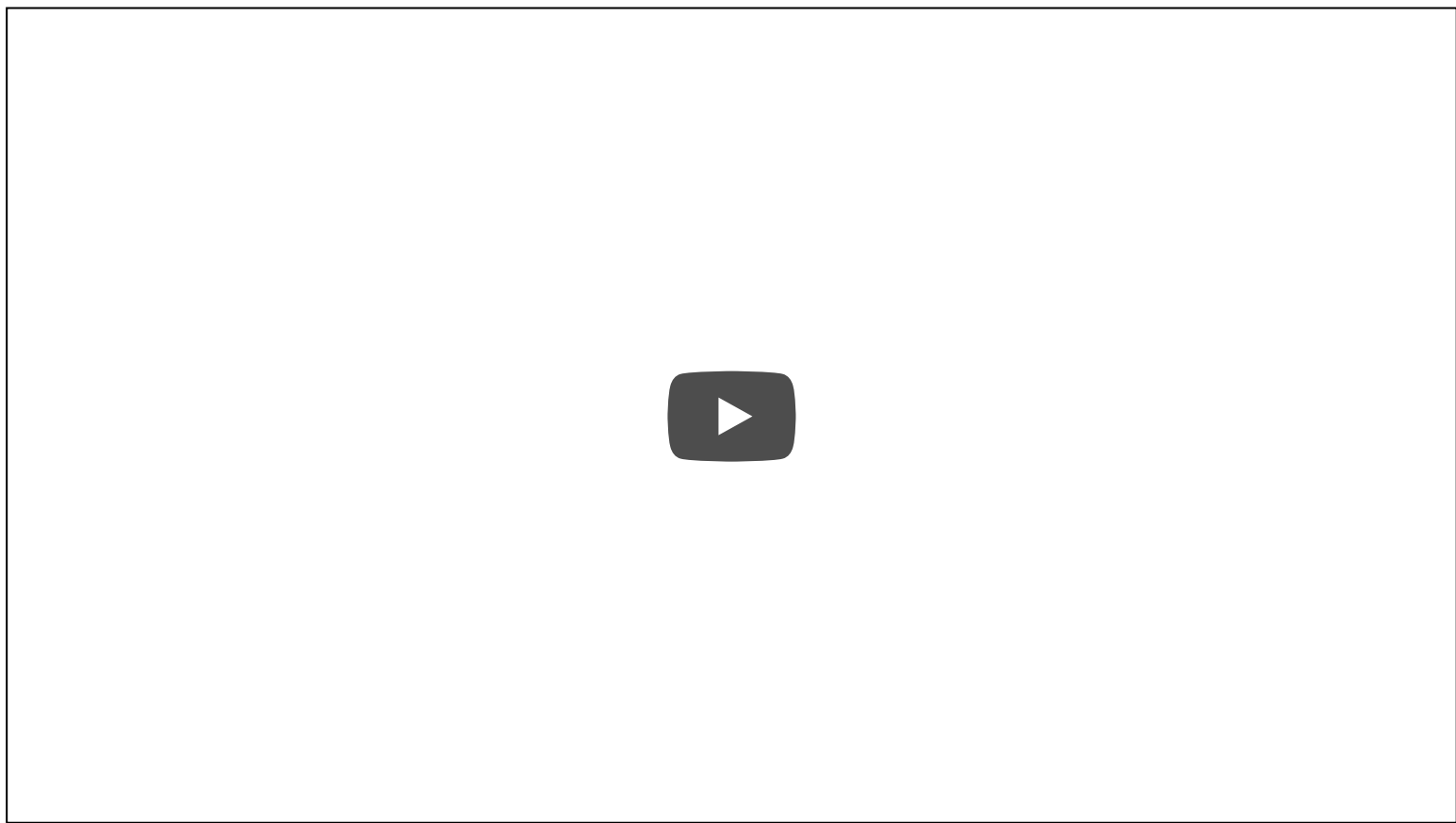
In this demonstration we're going to synchronize what you've learned about software development practices and machine learning, using AWS DeepComposer to explore those best practices against a real life use case.

Coding Along With The Instructor (Optional)

To create the custom GAN, you will need to use an instance type that is not covered in the Amazon SageMaker free tier. If you want to code along with the demo and build a custom GAN, you may incur a cost.

You can learn more about SageMaker costs in the [Amazon SageMaker pricing documentation](#)

Getting Started

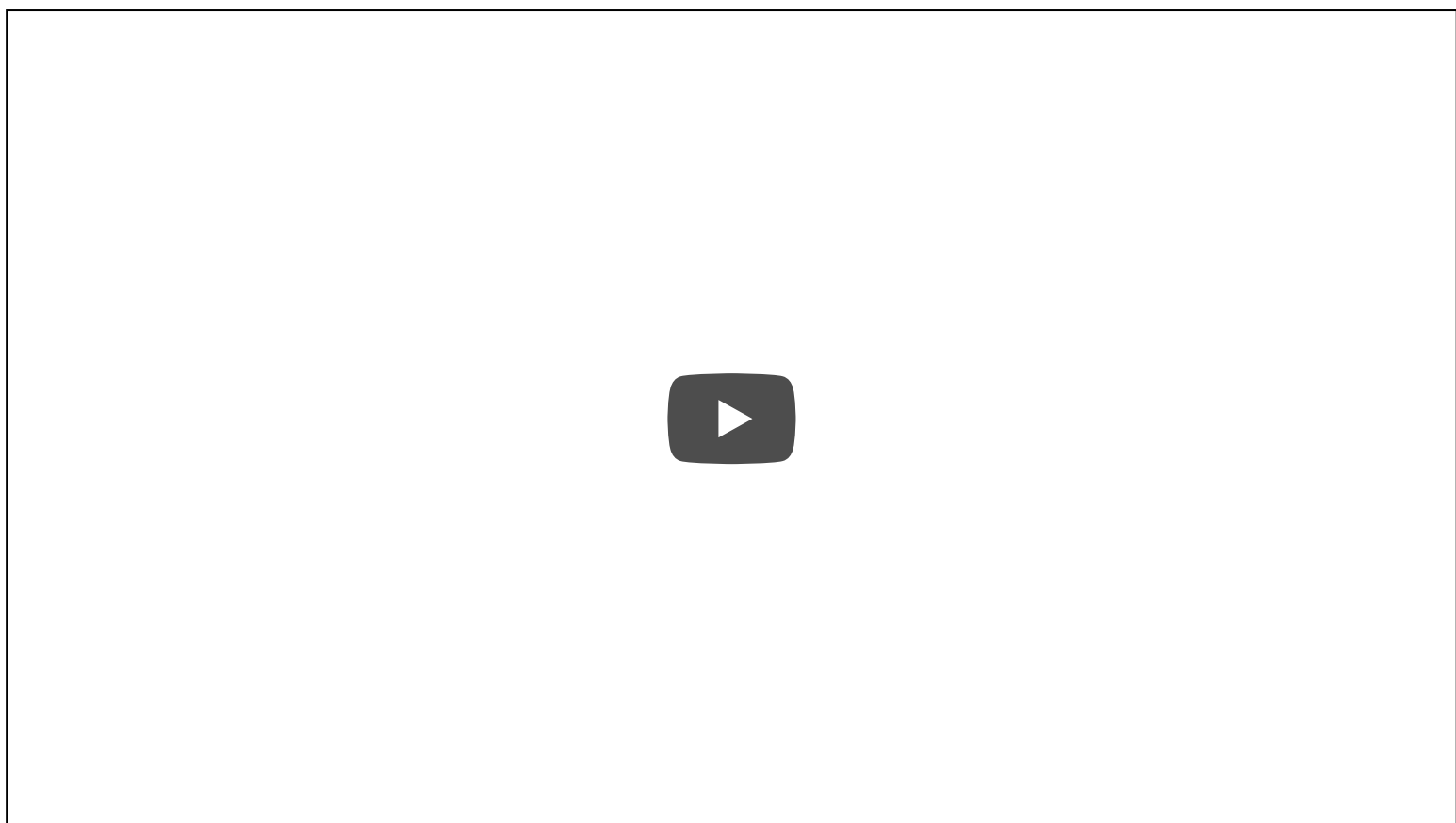


Setting Up the DeepComposer Notebook

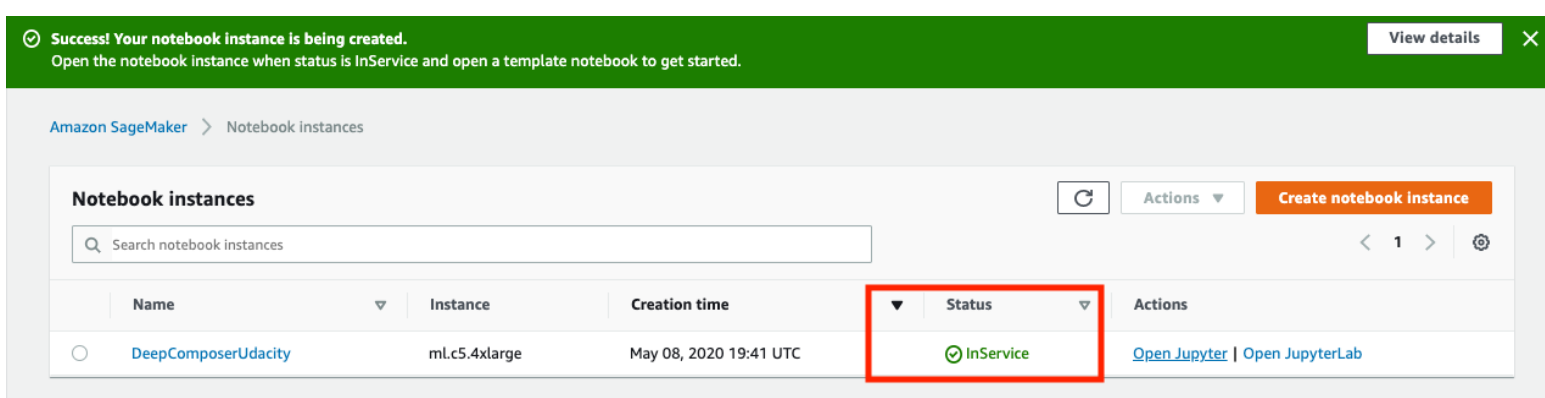
1. To get to the main Amazon SageMaker service screen, navigate to the AWS SageMaker console. You can also get there from within the AWS Management Console by searching for *Amazon SageMaker*.
2. Once inside the SageMaker console, look to the left hand menu and select **Notebook Instances**.
3. Next, click on **Create notebook instance**.
4. In the Notebook instance setting section, give the notebook a name, for example, `DeepComposerUdacity`.
5. Based on the kind of CPU, GPU and memory you need the next step is to select an instance type. For our purposes, we'll configure a `ml.c5.4xlarge`.
6. Leave the **Elastic Inference** defaulted to `none`.
7. In the Permissions and encryption section, create a new IAM role using all of the defaults.
8. When you see that the role was created successfully, navigate down a little ways to the Git repositories section
9. Select **Clone a public Git repository to this notebook instance only**
10. Copy and paste the public URL into the Git repository URL section:  
`https://github.com/aws-samples/aws-deepcomposer-samples`
11. Select **Create notebook instance**
12. Give SageMaker a few minutes to provision the instance and clone the Git repository

Exploring the Notebook

Now that it's configured and ready to use, let's take a moment to investigate what's inside the notebook.



When the status reads "InService" you can open the Jupyter notebook.



Open the Notebook

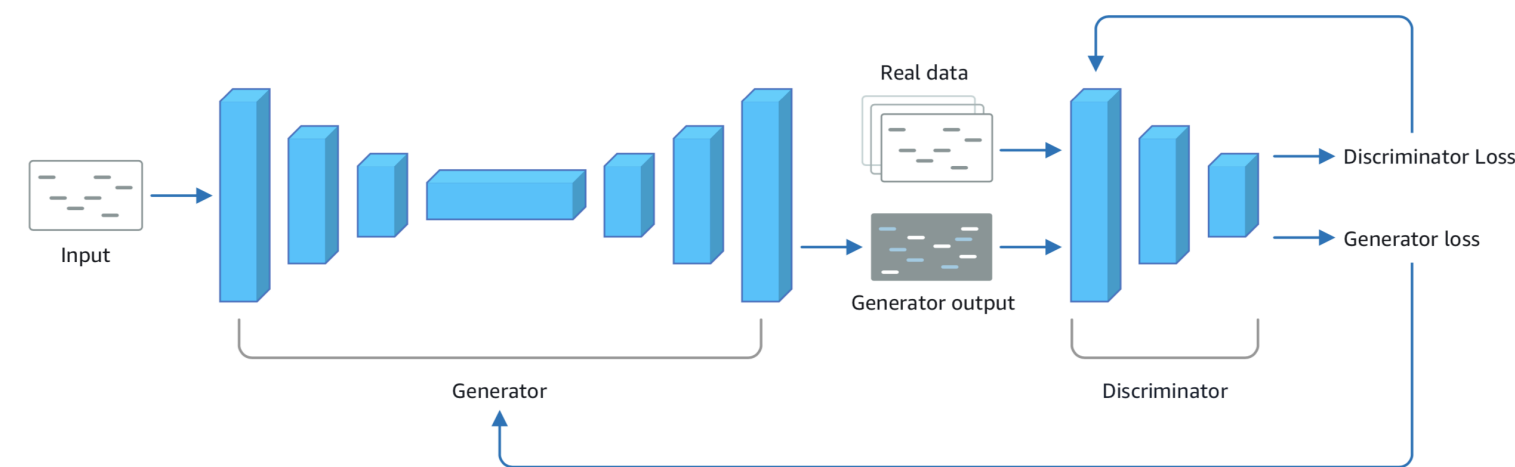
1. Click **Open Jupyter**.
2. When the notebook opens, click on **Lab 2**.
3. When the lab opens click on **GAN.ipynb**.

Review: Generative Adversarial Networks (GANs).

GANs consist of two networks constantly competing with each other:

- **Generator network** that tries to generate data based on the data it was trained on.
- **Discriminator network** that is trained to differentiate between real data and data which is created by the generator.

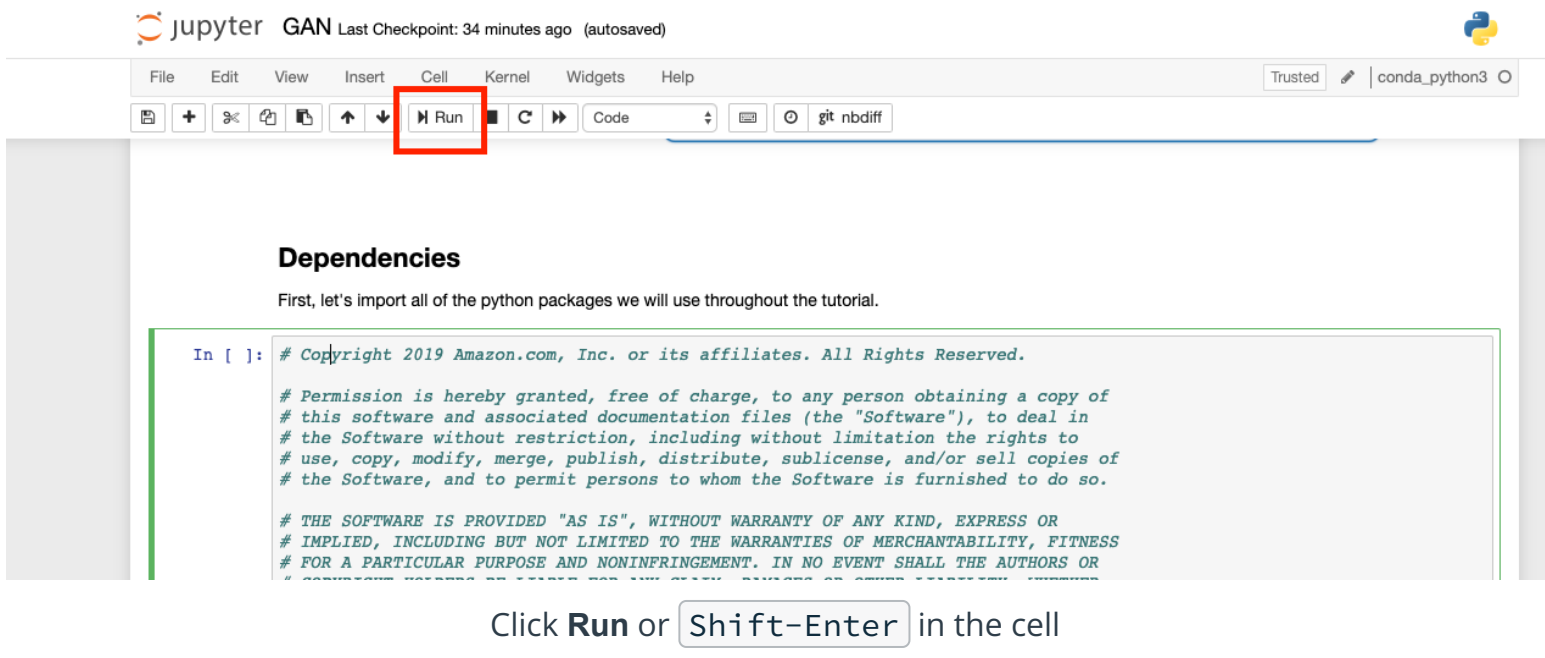
*Note:* The demo often refers to the **discriminator** as the **critic**. The two terms can be used interchangeably.



Set Up the Project

1. Run the first **Dependencies** cell to install the required packages
2. Run the second **Dependencies** cell to import the dependencies
3. Run the **Configuration** cell to define the configuration variables

*Note:* While executing the cell that installs dependency packages, you may see warning messages indicating that later versions of conda are available for certain packages. It is completely OK to ignore this message. It should not affect the execution of this notebook.

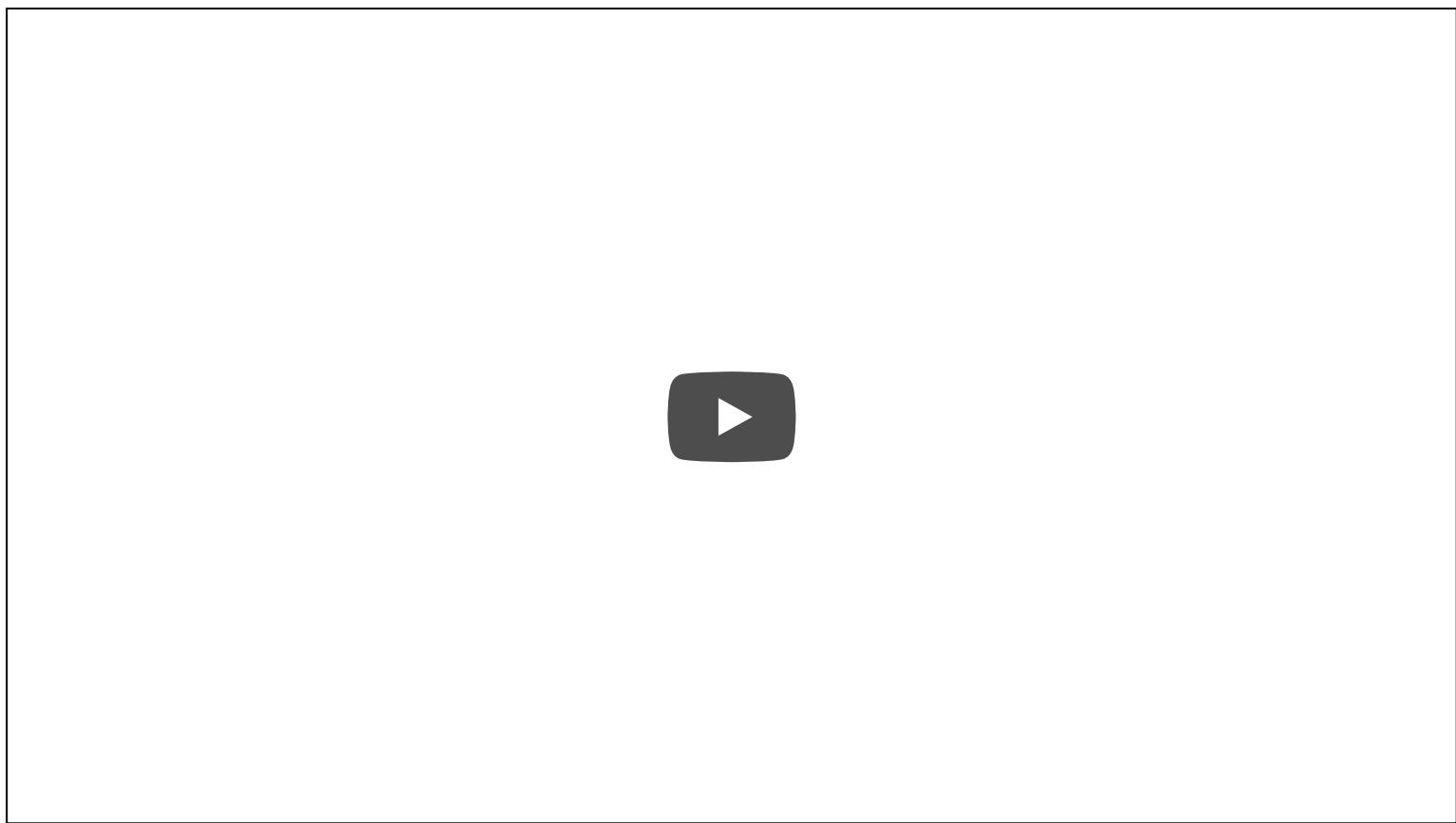


Good Coding Practices

- Do not hard-code configuration variables
- Move configuration variables to a separate `config` file
- Use code comments to allow for easy code collaboration

Data Preparation

The next section of the notebook is where we'll prepare the data so it can train the generator network.



Why Do We Need to Prepare Data?

Data often comes from many places (like a website, IoT sensors, or a hard drive, or physical paper) and it's usually not clean or in the same format. Before you can better understand your data, you need to