**Bekerja dengan Komputasi di Azure Machine Learning**

Salah satu manfaat utama cloud adalah kemampuan untuk menggunakan sumber daya komputasi sesuai permintaan yang dapat diskalakan untuk pemrosesan data besar yang hemat biaya. Dalam modul ini, Anda akan mempelajari bagaimana cara menggunakan komputasi awan di Azure Machine Learning untuk menjalankan eksperimen pelatihan dalam skala.

**Tujuan pembelajaran**

* Bekerja dengan lingkungan
* Bekerja dengan target komputasi

1. **Pengantar**

Dalam Azure Machine Learning, ilmuwan data dapat menjalankan eksperimen berdasarkan skrip yang memproses data, melatih model pembelajaran mesin, dan melakukan tugas ilmu data lainnya. Konteks runtime untuk setiap eksperimen yang dijalankan terdiri dari dua elemen:

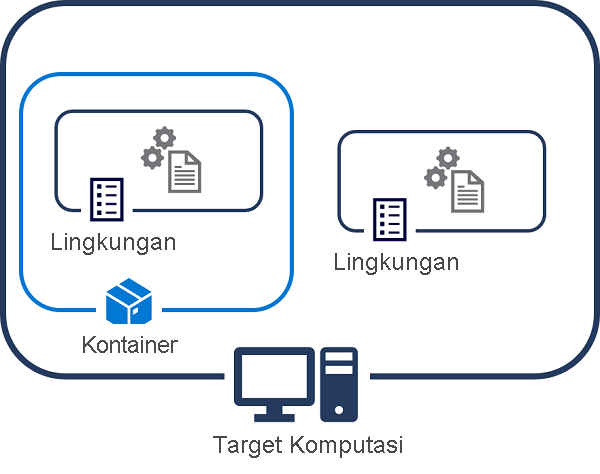
* Lingkungan untuk skrip, yang mencakup semua paket dimana tempat skrip bergantung.
* Target komputasi dimana tempat lingkungan akan disebarkan dan skrip dijalankan.

1. **Tujuan pembelajaran**

Dalam modul ini, Anda akan mempelajari cara:

* Membuat dan menggunakan lingkungan.
* Membuat dan menggunakan target komputasi.

1. **Pengenalan kepada lingkungan**



Kode Python berjalan dalam konteks lingkungan virtual yang menentukan versi runtime Python yang akan digunakan serta paket terinstal yang tersedia untuk kode tersebut. Di sebagian besar pemasangan Python, paket dipasang dan dikelola di lingkungan menggunakan **Conda** atau **pip**.

Untuk meningkatkan portabilitas, kami biasanya membuat lingkungan dalam kontainer docker yang nantinya akan dihosting di target komputasi, seperti komputer pengembangan, komputer virtual, atau kluster Anda di cloud.

1. **Lingkungan dalam Azure Machine Learning**

Secara umum, Azure Machine Learning menangani pembuatan lingkungan dan penginstalan paket untuk Anda - biasanya melalui pembuatan kontainer Docker. Anda dapat menentukan paket Conda atau pip yang Anda butuhkan, dan Azure Machine Learning akan membuat lingkungan untuk eksperimen.

Dalam solusi pembelajaran mesin perusahaan, di mana eksperimen dapat dijalankan dalam berbagai konteks komputasi, penting untuk mengetahui lingkungan tempat kode eksperimen Anda berjalan. Lingkungan dienkapsulasi oleh kelas **Lingkungan**; yang dapat Anda gunakan untuk membuat lingkungan dan menentukan konfigurasi runtime untuk eksperimen.

Anda dapat meminta Azure Machine Learning untuk mengelola pembuatan lingkungan dan pemasangan paket untuk menentukan lingkungan, lalu mendaftarkannya untuk digunakan kembali. Alternatifnya, Anda dapat mengelola lingkungan Anda sendiri dan mendaftarkannya. Hal ini memungkinkan untuk menentukan konteks runtime yang konsisten dan dapat digunakan kembali untuk eksperimen Anda - di mana pun skrip eksperimen dijalankan.

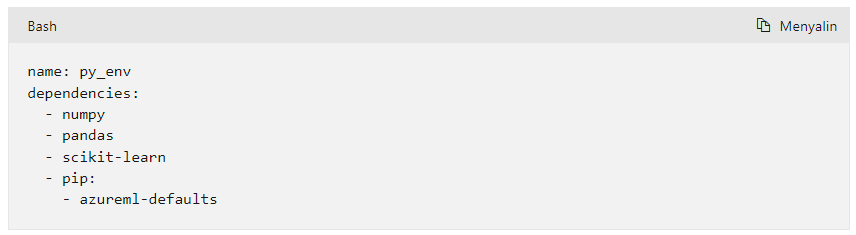
1. **Menciptakan lingkungan**

Ada beberapa cara untuk membuat lingkungan di Azure Machine Learning.

* **Membuat lingkungan dari file spesifikasi**

Anda dapat menggunakan file spesifikasi Conda atau pip untuk menentukan paket yang diperlukan dalam lingkungan Python, dan menggunakannya untuk membuat objek **Environment**.

Misalnya, Anda dapat menyimpan pengaturan konfigurasi Conda berikut dalam file bernama **conda.yml**:

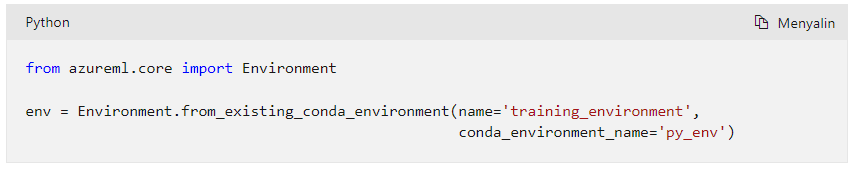


Anda kemudian dapat menggunakan kode berikut untuk membuat lingkungan Azure Machine Learning dari file spesifikasi yang sebelumnya disimpan:



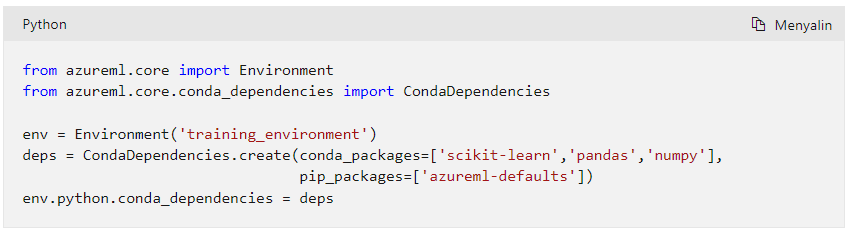
* **Menciptakan lingkungan dari lingkungan Conda sudah yang ada**

Jikalau anda telah memiliki lingkungan conda yang telah ditentukan di stasiun kerja, gunakan hal itu untuk menentukan lingkungan Azure Machine Learning:



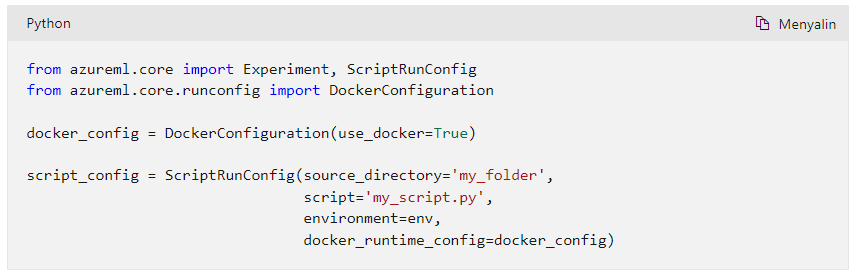
* **Membuat lingkungan dengan menentukan paket**

Anda dapat menentukan lingkungan dengan menentukan paket Conda dan pip yang Anda butuhkan dalam objek **CondaDependencies**, seperti ini:



1. **Mengonfigurasi kontainer lingkungan**

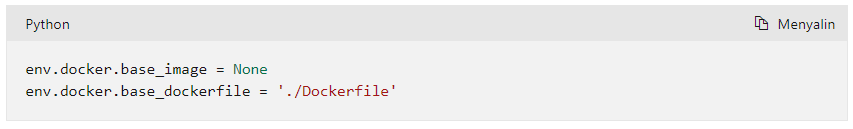
Biasanya, lingkungan untuk skrip eksperimen dibuat dalam kontainer. Kode berikut mengonfigurasi eksperimen berbasis skrip untuk menghosting lingkungan **env** yang dibuat sebelumnya di kontainer (ini adalah default kecuali Anda menggunakan **DockerConfiguration** dengan atribut **use\_docker** dari **Salah**, dalam hal ini lingkungan dibuat langsung di target komputasi).



Azure Machine Learning menggunakan pustaka gambar dasar sebagai kontainer, memilih basis yang sesuai untuk target komputasi yang Anda tentukan (misalnya, meliputi dukungan Cuda untuk komputasi berbasis GPU). Jika Anda telah membuat gambar kontainer kustom dan mendaftarkannya di registri penampung, Anda dapat mengambil alih gambar dasar default dan menggunakan milik Anda sendiri dengan memodifikasi atribut lingkungan properti **docker**.



Alternatifnya, Anda dapat membuat gambar yang sesuai permintaan berdasarkan gambar dasar dan pengaturan tambahan di file docker.



Secara default, Azure machine Learning menangani jalur Python dan dependensi paket. Jika gambar Anda sudah menyertakan penginstalan Python dengan dependensi yang Anda butuhkan, Anda dapat mengambil alih perilaku ini dengan mengatur **python.user\_managed\_dependencies** ke **Benar** dan menyetel jalur Python eksplisit untuk pamasangan Anda.



1. **Mendaftarkan dan menggunakan kembali lingkungan**

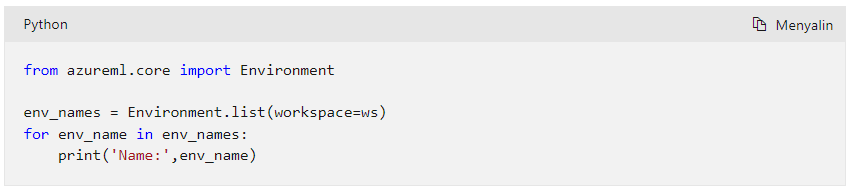
Setelah Anda membuat lingkungan, Anda dapat mendaftarkannya di ruang kerja dan dapat menggunakannya kembali untuk eksperimen mendatang yang memiliki dependensi Python yang sama.

* **Mendaftarkan lingkungan**

Gunakan metode **daftar** objek **Lingkungan** untuk mendaftarkan lingkungan:



Anda dapat menampilkan lingkungan yang terdaftar di ruang kerja Anda seperti ini:



1. **Mengambil dan menggunakan lingkungan**

Anda dapat mengambil lingkungan terdaftar dengan menggunakan metode **dapatkan** dari kelas **Lingkungan**, lalu menetapkannya ke **ScriptRunConfig**.

Misalnya, sampel kode berikut telah mengambil lingkungan yang terdaftar training\_environment, dan menetapkannya ke konfigurasi eksekusi skrip:



Ketika eksperimen berdasarkan penaksir dijalankan, Azure Machine Learning akan mencari lingkungan yang sudah ada dan cocok dengan definisi, dan jika tidak ada yang ditemukan, lingkungan baru akan dibuat berdasarkan spesifikasi lingkungan yang terdaftar.

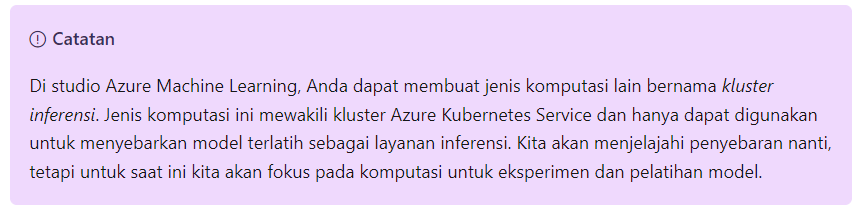
1. **Pengenalan target komputasi**

Dalam Azure Machine Learning, Target Komputasi adalah komputer fisik atau virtual tempat eksperimen dijalankan.

1. **Jenis Komputasi**

Azure Machine Learning mendukung berbagai jenis komputasi untuk eksperimen dan pelatihan. Hal ini memungkinkan Anda memilih jenis target komputasi yang paling sesuai untuk kebutuhan tertentu Anda.

* **Komputasi lokal** - Anda dapat menentukan target komputasi lokal untuk sebagian besar tugas pemrosesan di Azure Machine Learning. Hal ini dapat menjalankan eksperimen pada target komputasi yang sama dengan kode yang digunakan untuk memulai eksperimen, yang mungkin saja merupakan stasiun kerja fisik Anda atau komputer virtual seperti Azure Machine Learninginstans komputasi tempat dimana Anda menjalankan buku catatan. Komputasi lokal umumnya merupakan pilihan tepat selama pengembangan dan pengujian dengan volume data rendah hingga sedang.
* **Kluster komputasi** - Untuk beban kerja eksperimen dengan persyaratan skalabilitas tinggi, Anda dapat menggunakan kluster komputasi Azure Machine Learning; yang merupakan kluster multi sampul dari Virtual Machines yang secara otomatis meningkatkan atau menurunkan untuk memenuhi permintaan. Ini adalah cara hemat biaya untuk menjalankan eksperimen yang perlu menangani volume data yang besar atau menggunakan pemrosesan paralel untuk mendistribusikan beban kerja dan juga mengurangi waktu yang diperlukan untuk menjalankannya.
* **Komputasi terlampir** - Jika Anda sudah menggunakan lingkungan komputasi berbasis Azure untuk ilmu data, seperti komputer virtual atau kluster Azure Databricks, Anda dapat melampirkannya ke ruang kerja Azure Machine Learning dan menggunakannya sebagai komputasi target untuk jenis beban kerja tertentu.



Kemampuan untuk menetapkan eksperimen yang berjalan di target komputasi tertentu membantu Anda menerapkan ekosistem ilmu data yang fleksibel dengan cara berikut:

* Kode dapat dikembangkan dan diuji pada komputasi lokal atau yang berbiaya rendah, dan kemudian dipindahkan ke komputasi yang lebih skalabel untuk beban kerja produksi.
* Anda dapat menjalankan proses individual pada target komputasi yang paling sesuai dengan kebutuhannya. Misalnya, dengan menggunakan komputasi berbasis GPU untuk melatih model pembelajaran mendalam, dan beralih ke komputasi khusus CPU berbiaya lebih rendah untuk menguji dan mendaftarkan model yang telah dilatih.

Salah satu manfaat inti dari komputasi awan adalah kemampuan untuk mengelola biaya dengan membayar hanya untuk apa yang Anda gunakan. Di Azure Machine Learning, Anda dapat memanfaatkan prinsip ini dengan menentukan target komputasi yang:

* Bisa langsung memulai sesuai permintaan dan berhenti secara otomatis saat tidak lagi diperlukan.
* Berskala secara otomatis berdasarkan kebutuhan pemrosesan beban kerja.

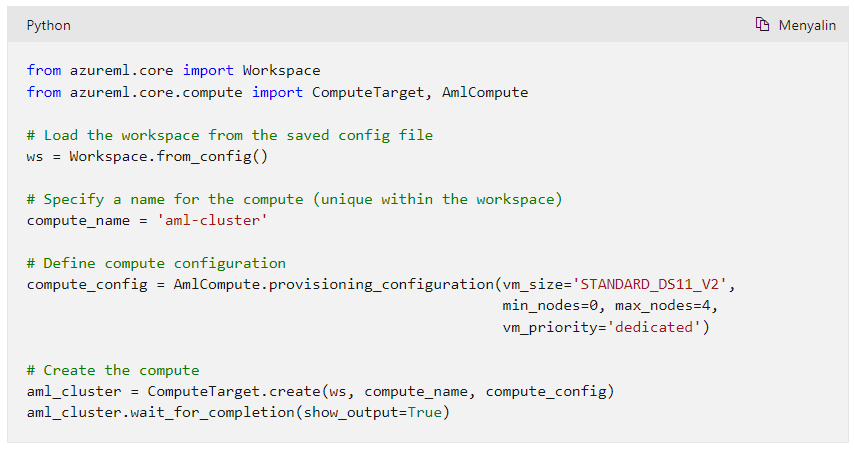
1. **Membuat target komputasi**

Cara paling umum untuk membuat atau melampirkan target komputasi adalah dengan menggunakan halaman **Komputasi** di studio Azure Machine Learning, atau menggunakan SDK Azure Machine Learning untuk menyediakan target komputasi dalam kode.

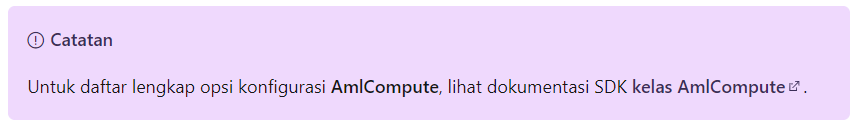
1. **Membuat target komputasi terkelola dengan SDK**

terkelola Target komputasi adalah target yang dikelola oleh Azure Machine Learning, seperti kluster komputasi Azure Machine Learning.

Untuk membuat kluster komputasi Azure Machine Learning, gunakan kelas **azureml.core.compute.ComputeTarget** dan kelas **AmlCompute**, seperti ini:



Dalam contoh berikut, sebuah kluster yang mencapai empat simpul dan didasarkan pada gambar komputer virtual STANDARD\_DS12\_v2 akan dibuat. Prioritas untuk komputer virtual (VM) telah di set ke dedikasi, artinya komputer tersebut dicadangkan untuk digunakan dalam kluster ini (alternatifnya adalah menentukan prioritas rendah, yang memiliki biaya lebih rendah tetapi berarti VM dapat didahulukan jika beban kerja dengan prioritas lebih tinggi memerlukan komputasi).

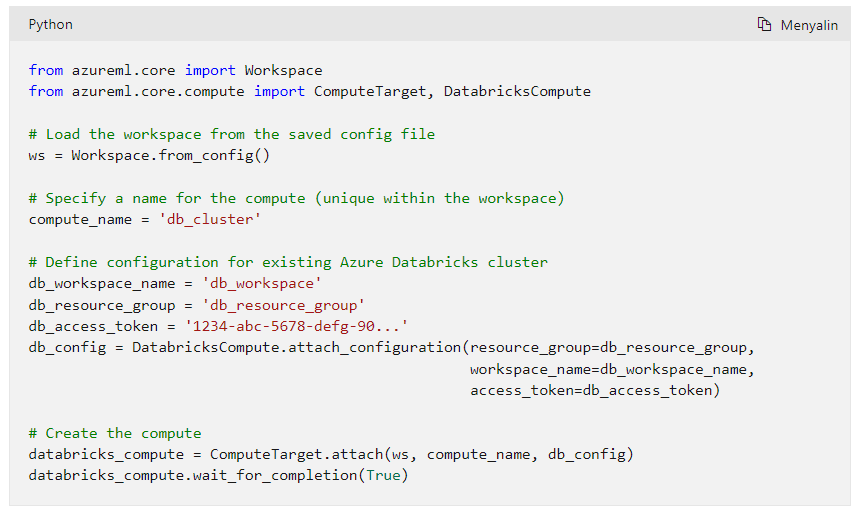


1. **Melampirkan target komputasi yang tidak dikelola dengan SDK**

tidak terkelola Target komputasi adalah yang ditentukan dan dikelola di luar ruang kerja Azure Machine Learning; misalnya, komputer virtual Azure atau kluster Azure Databricks.

kode untuk melampirkan ke target komputasi yang sudah ada namun belum terkelola mirip dengan kode yang digunakan untuk membuat target komputasi terkelola, kecuali bahwa Anda harus menggunakan metode **ComputeTarget.attach()** untuk melampirkan komputasi yang ada berdasarkan target pengaturan konfigurasi tertentu.

Misalnya, kode berikut ini dapat digunakan untuk melampirkan kluster Azure Databricks yang sudah ada:



1. **Memeriksa target komputasi yang ada**

Dalam banyak kasus, Anda ingin memeriksa keberadaan target komputasi, dan hanya membuat target baru jika belum ada target komputasi dengan nama yang ditentukan. Untuk melakukannya, Anda dapat mengambil **ComputeTargetException** pengecualian, seperti ini:



Informasi Lebih Lanjut: Untuk informasi selengkapnya tentang membuat target komputasi, lihat Menyiapkan dan menggunakan target komputasi untuk pelatihan model : <https://learn.microsoft.com/id-id/azure/machine-learning/v1/how-to-set-up-training-targets>

dalam dokumentasi Azure Machine Learning.

1. **Gunakan target komputasi**

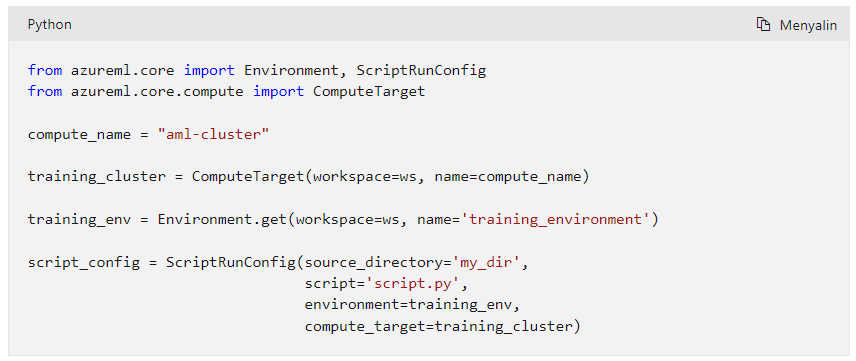
Setelah membuat atau melampirkan target komputasi di ruang kerja, Anda dapat menggunakannya untuk menjalankan beban kerja tertentu; seperti eksperimen.

Untuk menggunakan target komputasi tertentu, Anda dapat menentukannya dalam parameter yang sesuai untuk konfigurasi atau penaksir percobaan. Misalnya, kode berikut mengonfigurasi penaksir untuk menggunakan target komputasi bernama aml-cluster:



Ketika eksperimen dikirimkan, proses akan diantrekan saat aml-cluster target komputasi dimulai dan lingkungan yang ditentukan dibuat di dalamnya, lalu proses tersebut akan diproses di lingkungan komputasi.

Alih-alih menentukan nama target komputasi, Anda dapat menentukan objek **ComputeTarget**, seperti ini:



1. **Latihan - Bekerja dengan Konteks Komputasi**

Sekarang adalah kesempatan Anda untuk bekerja dengan lingkungan dan target komputasi di Azure Machine Learning.

Dalam latihan ini, Anda akan:

* Membuat dan menggunakan lingkungan.
* Membuat dan menggunakan target komputasi.

1. **Instruksi**

Ikuti petunjuk ini untuk menyelesaikan latihan.

1. Jika Anda belum memiliki langganan Azure, daftar untuk uji coba gratis di https://azure.microsoft.com.
2. Menampilkan repo latihan di <https://aka.ms/mslearn-dp100.>

Link : <https://microsoftlearning.github.io/mslearn-dp100/>

1. Jika Anda belum melakukannya, selesaikan latihan Membuat ruang kerja Azure Machine Learning untuk menyediakan ruang kerja Azure Machine Learning, membuat instans komputasi, dan klon file yang diperlukan.
2. Selesaikan latihan Bekerja dengan komputasi.
3. **Ringkasan**

Dalam modul ini, Anda belajar cara:

* Membuat dan menggunakan lingkungan.
* Membuat dan menggunakan target komputasi.

Untuk informasi selengkapnya tentang lingkungan di Azure Machine Learning, lihat Menggunakan kembali lingkungan untuk pelatihan dan penerapan dengan menggunakan Azure Machine Learning.

Link : <https://learn.microsoft.com/id-id/azure/machine-learning/how-to-manage-environments-v2?tabs=cli>

Untuk informasi selengkapnya tentang target komputasi di Azure Machine Learning, lihat Apa yang dimaksud dengan target komputasi di Azure Machine Learning?

Link : <https://learn.microsoft.com/id-id/azure/machine-learning/concept-compute-target>