Get the best out of Live Sessions HOW?

Check your Internet Connection

Log in 10 mins before, and check your internet connection to avoid any network issues during the LIVE session.

Speak with the Instructor

By default, you will be on mute to avoid any background noise. However, if required you will be **unmuted by instructor**.

Clear Your Doubts

Feel free to clear your doubts. Use the "**Questions**" tab on your webinar tool to interact with the instructor at any point during the class.

Let us know if you liked our content

Please share feedback after each class. It will help us to enhance your learning experience.

edureka!

Microsoft Azure Developer Associate (AZ-204)

COURSE OUTLINE
MODULE 02

Introduction To Azure IaaS Compute Solutions

Implementing Azure Batch Service & Disk Encryption

Designing & Developing Applications That Use Containers

Implementing Azure App Service Web Apps & Mobile Apps

Implementing Azure App Service API Apps & Azure Functions

Developing Solutions That Use Azure Table Storage & Cosmos DB

Developing Solutions That Use Relational Database & Azure Blob Storage

Implementing Authentication & Access Control In Azure

Implementing Secure Data Solutions And Integrate Caching & CDN

Instrument Monitoring, Logging & Scalability Of Apps & Services

Connecting To And Consuming Azure & Third-party Services

Developing Event-based & Message-based Solutions In Azure

Module 2 – Implementing Azure Batch Service and Disk Encryption

edureka!

Topics

- ARM templates for Azure Virtual Machines
- Azure Disk Encryption
- Data Privacy And Data Sovereignty In Azure
- Encryption Scenarios
- Encryption Workflow
- Azure Batch Services
- Intrinsically Parallel Workloads
- Batch Capabilities
- Azure Batch Workflow
- Azure Accounts For Batch Development
- Batch Service And Management APIs
- Batch Command-line Tools

Objectives

After completing this module, you should be able to:

- Create and deploy ARM templates by using the Azure Portal and Visual Studio
- Configure Azure Disk Encryption for VMs
- Run batch jobs by using different methods
- Manage batch jobs by using the Batch Service API

Templates in Azure Resource Manager

edureka!

An ARM Template Syntax

- A template usually consists of the following elements

```
{
"$schema": "http://schema.management.azure.com/schemas/2018-01-01/deploymentTemplate.json#",
"contentVersion": "",
"parameters": { },
"variables": { },
"functions": { },
"resources": [ ],
"outputs": { }
}
```

Expressions are written within **JSON** string literals whose first and last characters are the brackets: [and], respectively

- Templates provide a method of deploying resources to Azure in a declarative fashion

- The value of the expression is evaluated when the template is deployed

Template Syntax

Element name	Required	Description
$schema	Yes	Location of the JSON schema file that describes the version of the template language
contentVersion	Yes	Version of the template
parameters	No	Values that are provided when deployment is executed to customize resource deployment
variables	No	Values that are used as JSON fragments in the template to simplify template language expressions
functions	No	Values that are used as JSON fragments in the template to simplify template language expressions
resources	Yes	Resource types that are deployed or updated in a resource group
outputs	No	Values that are returned after deployment

Template Syntax – Parameters

- The **parameters** section of the template contains values which are input when deploying the resources
- These parameter values enable customization of the deployment since they are suited for a particular environment (such as dev, test, and production).
- Parameters are not mandatory, but without parameters your template would always deploy the same resources with the same names, locations, and properties.

```
"parameters": {
  "siteNamePrefix": {
    "type": "string",
    "metadata": {
      "description": "The name prefix of the web app that you wish to create."
    }
  },
},
```

Template Syntax – Variables

- The **variables** section contain values that can be used throughout your template
- They often simplify your template by reducing complex expressions

```
"variables": {
  "webSiteName": "[concat(parameters('siteNamePrefix'), uniqueString(resourceGroup().id))]",
}
```

Template Syntax – Functions

- Functions are typically define complicated expressions that that used multiple times within the template
- User-defined functions may be created from expressions and functions that are supported in templates
- User-defined functions have a few restrictions placed on their use:

01 The functions can't access variables

02 The function can't use the reference functions

03 Parameters for the function cannot have default values

Template Syntax – Functions (Cont.)

- Here is an example of a function used in the ARM template

```
"functions": [
  {
    "namespace": "contoso",
    "members": {
      "uniqueName": {
        "parameters": [
          {
            "name": "namePrefix",
            "type": "string"
          }
        ],
        "output": {
          "type": "string",
          "value": "[concat(toLower(parameters('namePrefix')),
uniqueString(resourceGroup().id))]"
        }}}}],
```

Template Syntax – Functions (Cont.)

- Here is how a function call happens in Azure templates

```
"resources": [
  {
    "name": "[contoso.uniqueName(parameters('storageNamePrefix'))]",
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2016-01-01",
    "sku": {
      "name": "Standard_LRS"
    },
    "kind": "Storage",
    "location": "South Central US",
    "tags": {},
    "properties": {}
  }
]
```

Template Functions – Resources

- In the resources section, the resources that are to be deployed or updated are defined

```
"resources": [
  {
    "name": "[contoso.uniqueName(parameters('storageNamePrefix'))]",
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2016-01-01",
    "sku": {
      "name": "Standard_LRS"
    },
    "kind": "Storage",
    "location": "South Central US",
    "tags": {},
    "properties": {}
  }
]
```

Template Functions – Outputs

- In the Outputs section the values that are returned from deployment
- For example, you could return the URI to access a deployed resource.

```
"outputs": {
  "newHostName": {
    "type": "string",
    "value": "[reference(variables('webSiteName')).defaultHostName]"
  }
}
```

Create a Windows VM from an ARM Template

- The upcoming demo will show you how to create a **Windows** Virtual Machine by using an Azure Resource Manager **template** and Azure **PowerShell** from the Azure ***Cloud shell***
- The template used in this demo deploys a **single virtual machine** running Windows Server in a new **virtual network** with a **single subnet**
- For creating a Linux virtual machine, you can check here
- Creating an Azure virtual machine usually includes two steps:
  - Create a resource group
  - Create a virtual machine
- The following demo creates a VM from an Azure Quickstart template

Demo 1 – Create a Windows VM from an ARM Template

Refer demo doc-1

Azure Disk Encryption

edureka!

Data Privacy and Data Sovereignty in Azure

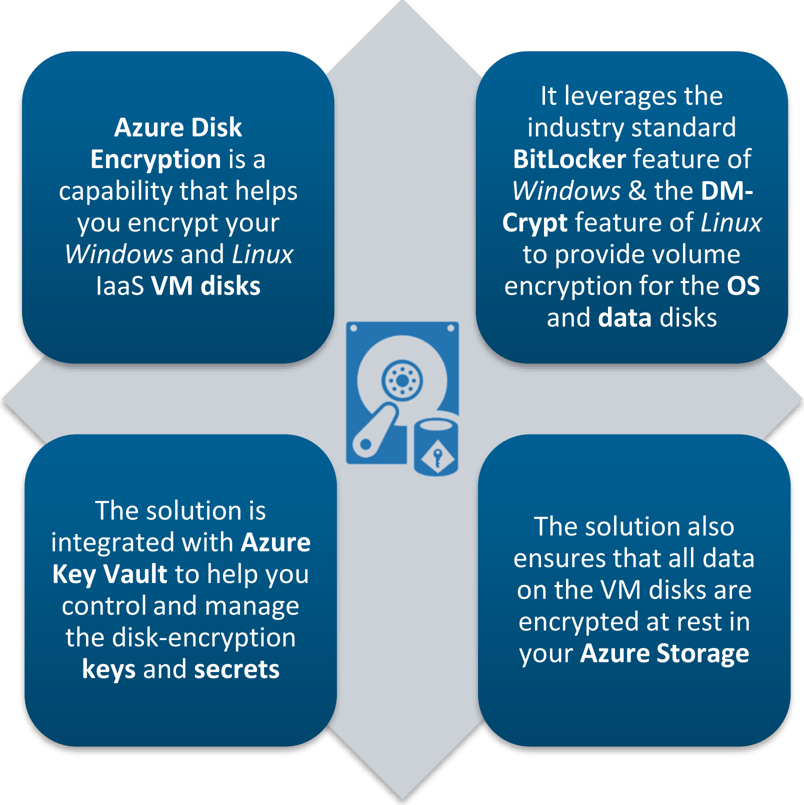
Azure enables you to control your **Azure-hosted data** through a range of advanced technologies to **encrypt**, **control** and **manage** encryption **keys**, and control and audit **access of data**

This control provides Azure customers with the ***flexibility*** to choose the solution that best meets their business needs

One such technology solution is "Azure Disk Encryption for Windows and Linux IaaS virtual machines (VMs).“

This technology helps **protect** and **safeguard** your data to meet your *organizational security* and *compliance commitments*

Azure Disk Encryption for IaaS VMs



Azure Disk Encryption is a capability that helps you encrypt your *Windows* and *Linux* IaaS **VM disks**

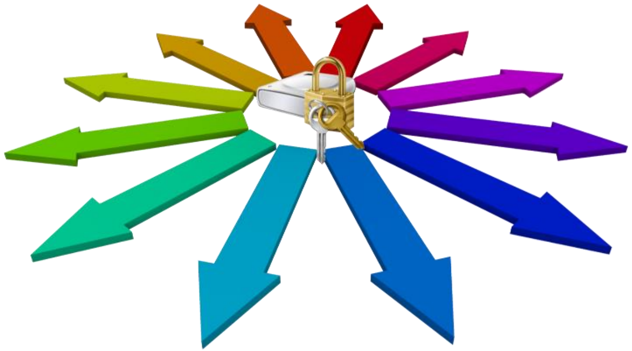
It leverages the industry standard **BitLocker** feature of *Windows* & the **DM-Crypt** feature of *Linux* to provide volume encryption for the **OS** and **data** disks

The solution is integrated with **Azure Key Vault** to help you control and manage the disk-encryption **keys** and **secrets**

The solution also ensures that all data on the VM disks are encrypted at rest in your **Azure Storage**

Encryption Scenarios

- The Disk Encryption solution supports the following customer scenarios:
  - Enable encryption on new Windows IaaS VMs created from pre-encrypted VHD and encryption keys
  - Enable encryption on new IaaS VMs created from the supported Azure Gallery images
  - Enable encryption on existing IaaS VMs that run in Azure
  - Enable encryption on Windows VM scale sets
  - Enable encryption on data drives for Linux VM scale sets
  - Enable encryption of managed disk VMs
  - Back up and restore of encrypted VMs



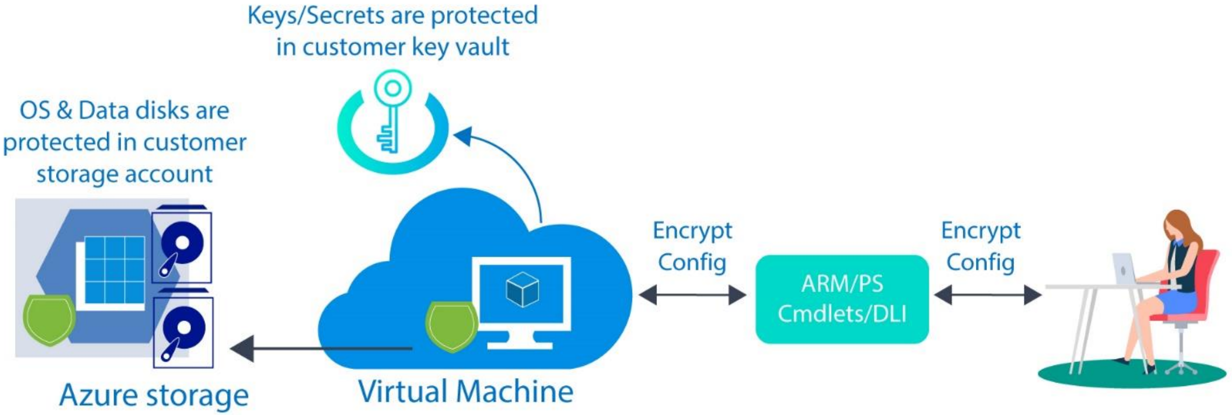
Encryption Scenarios – Unsupported

- The solution doesn't support the following scenarios, features, and technology:
  - Basic tier IaaS VMs
  - OS drive encryption for Linux VM scale sets
  - IaaS VMs that are created by using the classic VM creation method
  - Enable encryption of customer custom images on Linux IaaS VMs
  - Azure Files (shared file system)
  - Network File System (NFS)
  - Dynamic volumes

Encryption Workflow

- To enable disk encryption for Windows and Linux VMs, do the following steps:

    1. Choose an encryption scenario from the scenarios listed in the **Encryption scenarios** slide
    2. Opt in to enable disk encryption via the Azure Disk Encryption ARM **template**, PowerShell **cmdlets**, or the Azure **CLI**, and specify the encryption **configuration**
    3. **Grant access** to the Azure platform to read the encryption **key** material (**BitLocker** encryption keys for Windows systems and Passphrase for Linux) from your **key vault** to enable encryption on the IaaS VM
    4. Azure updates the VM service model with **encryption** and the key vault **configuration**, and sets up your encrypted **VM**


Keys/Secrets are protected in customer key vault
OS & Data disks are protected in customer storage account
Azure storage
Virtual Machine
Encrypt Config
ARM/PS Cmdlets/DLI
Encrypt Config

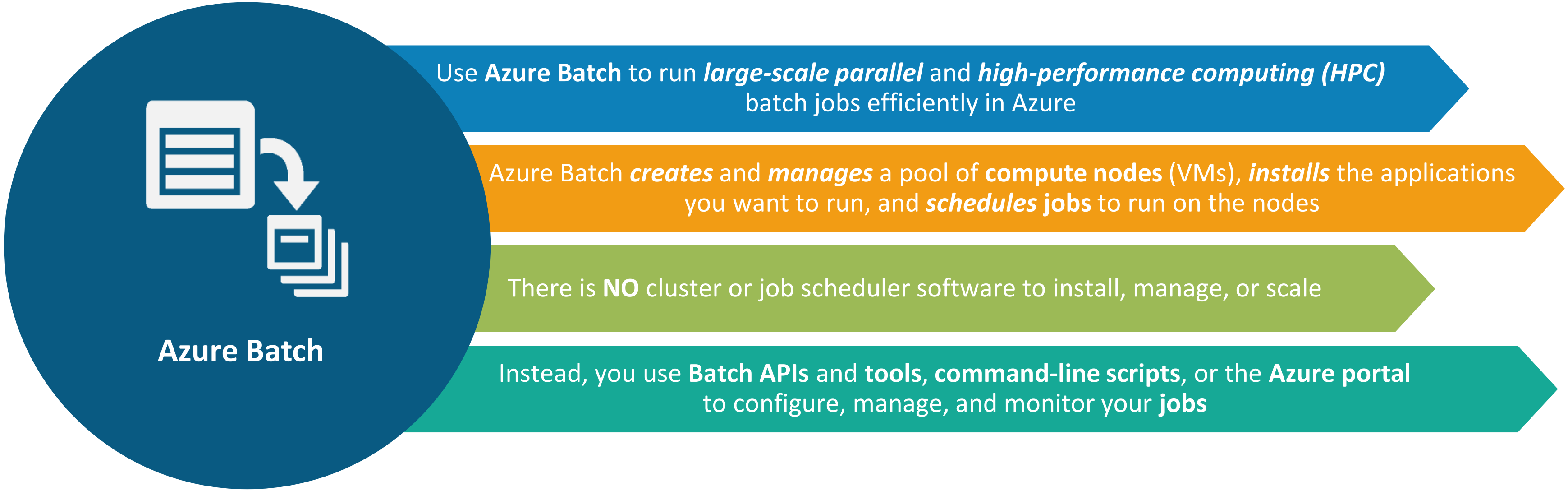
Demo 2 – Enable Azure Disk Encryption for Windows IaaS VMs

Refer demo doc-2

Azure Batch Services

edureka!

What is Azure Batch?



Azure Batch

Use **Azure Batch** to run ***large-scale parallel*** and ***high-performance computing (HPC)*** batch jobs efficiently in Azure

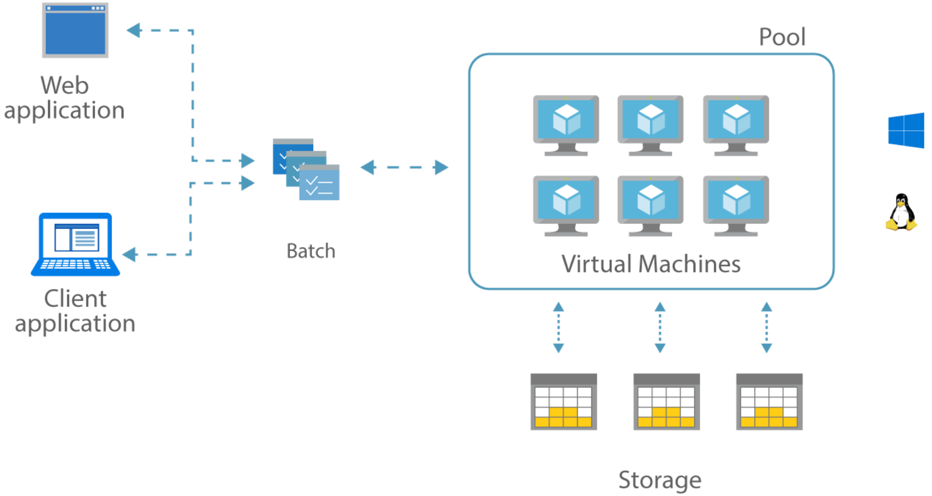
Azure Batch ***creates*** and ***manages*** a pool of **compute nodes** (VMs), ***installs*** the applications you want to run, and ***schedules*** **jobs** to run on the nodes

There is **NO** cluster or job scheduler software to install, manage, or scale

Instead, you use **Batch APIs** and **tools**, **command-line scripts**, or the **Azure portal** to configure, manage, and monitor your **jobs**

Azure Batch for Building SaaS Applications

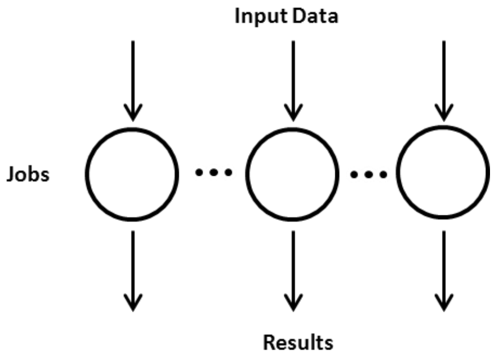
- **Developers** can use Batch as a ***platform service*** to build SaaS applications or client apps where ***large-scale execution*** is required
- For example, build a service with Batch to run a Monte Carlo risk simulation for a financial services company, or a service to process many images



__NOTE__: There is **NO** additional charge for using Batch. You **only** pay for the underlying resources consumed, such as the virtual machines, storage, and networking.

Run Parallel Workloads Using Azure Batch

- **Azure Batch** works well with **intrinsically parallel** (also known as "embarrassingly parallel") workloads
- Intrinsically parallel workloads are those where the applications can run **independently**, and each instance completes ***part of the work***

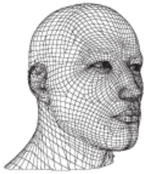


- When the applications are *executing*, they might access some **common data**, but they ***do not*** communicate with **other** instances of the application
- These workloads can therefore run at a **large scale**, determined by the ***amount of compute resources available*** to run applications *simultaneously*

Intrinsically Parallel Workloads – Examples

- Some examples of intrinsically parallel workloads you can bring to Batch:

Financial risk modeling using Monte Carlo simulations



VFX and 3D image rendering

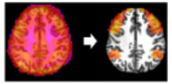
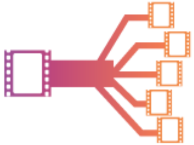


Image analysis & processing



Media transcoding

A T G G T C A T T G C G C C G A A T A G
T T C G G A T G G T C A T T G C G C C G

Genetic sequence analysis

Optical character recognition (OCR)

CRM
LOG
ERP
EXTRACT
TRANSFORM
LOAD

Data ingestion, processing, & ETL operations

Software test execution

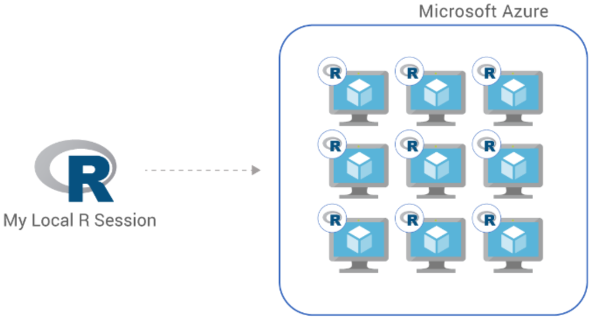
edureka!

Additional Batch Capabilities

Higher-level, workload-specific capabilities are also available for Azure Batch:

Batch supports large-scale rendering workloads with **rendering tools** including Autodesk Maya, 3ds Max, Arnold, and V-Ray.

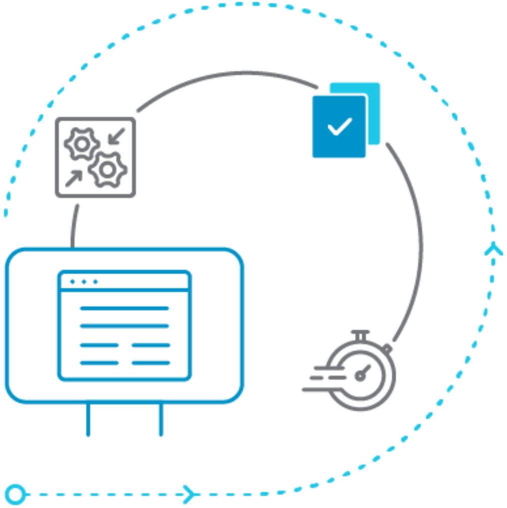
R users can install the **doAzureParallel R** package to easily scale out the execution of ***R algorithms*** on Batch pools.



edureka!

Azure Batch – How It Works?

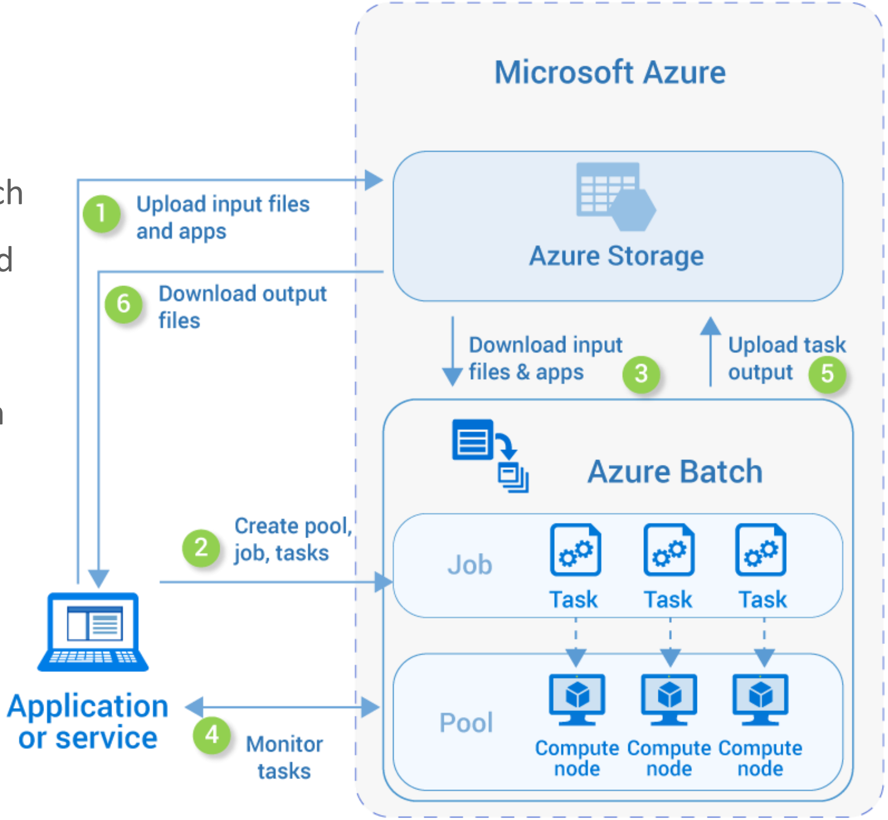
- A common scenario involves *scaling out intrinsically parallel work*, such as:
  - **rendering of images for 3D scenes**, on a ***pool*** of compute nodes
- This pool of compute nodes can be your "render farm" that provides up to thousands of cores to your **rendering job**
- In the next slide, let us checkout a scenario of a common Batch workflow:
  - with a client application or hosted service using Batch to run a parallel workload



edureka!

Using Batch to Run a Parallel Workload – Workflow

1. Upload input files and the applications to process those files to your **Azure Storage account**
2. Create a **Batch pool** of compute nodes in your Batch account, **a job** to run the workload on the pool, and **tasks** in the job
3. Download **input files** and the **applications** to Batch
4. **Monitor** task execution
5. Upload **task output**
6. Download **output files**



Azure Batch APIs and Tools

Overview of Batch APIs and Tools

- Processing parallel workloads with Azure Batch is done **programmatically** by using one of the **Batch APIs**
- Your client application *or* service can use the Batch APIs to **communicate** with the Batch service
- With the Batch APIs, you can **create** and **manage** pools of compute nodes, either **VMs** or **cloud services**
- You can then **schedule jobs** and **tasks** to run on those **nodes**

Azure Accounts for Batch Development

- When you develop Batch solutions, you use the following accounts in your Azure subscription:

Batch Account:

❑ Azure Batch **resources**, including pools, compute nodes, jobs, and tasks, are ***associated*** with an Azure Batch account

❑ When your application ***makes a request*** against the Batch service, it ***authenticates*** the request using the **Azure Batch account name**, the **URL** of the account, and either **an access key** or an **Azure Active Directory token**

Storage Account:

❑ Batch includes **built-in support** for working with ***files*** in Azure Storage

❑ Nearly every Batch scenario uses Blob storage for *staging* the **programs** that your ***tasks run,*** the **data** that they ***process***, and for the storage of **output data** that they generate

Batch Service and Management APIs

Batch Service APIs

- Batch REST – N/A
- Batch .NET – NuGet
- Batch Python – PyPI
- Batch Node.js – npm
- Batch Java – Maven

Batch Management APIs

- Batch Management REST – N/A
- Batch Management .NET – NuGet
- Batch Management Python – PyPI
- Batch Management Node.js – npm
- Batch Management Java – Maven

Click on the hyperlinks to check out the commands to install the respective Client library packages.

Batch Command-line Tools

- These command-line tools provide the same functionality as the Batch service and Batch Management APIs:

Batch PowerShell cmdlets:

The Azure Batch cmdlets in the Azure PowerShell module enable you to manage Batch resources with PowerShell.

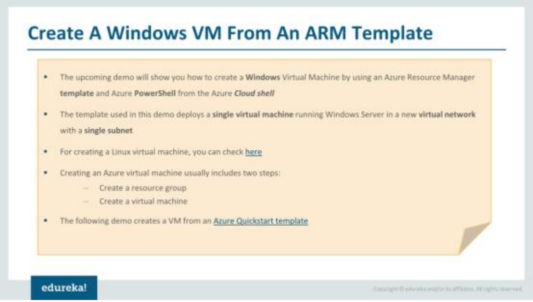
Azure CLI:

The Azure CLI is a cross-platform toolset that provides shell commands for interacting with many Azure services, including the Batch service and Batch Management service.

Demo 3 – Run a Batch Job by using Azure CLI, Azure Portal and C# Code

edureka!

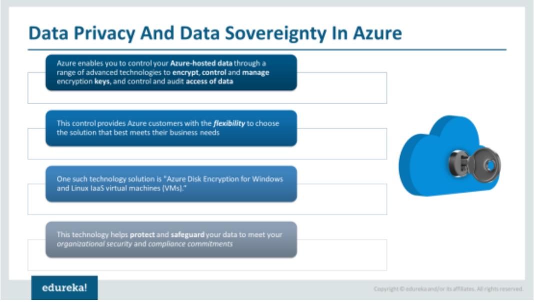
Summary



Create A Windows VM From An ARM Template

- The upcoming demo will show you how to create a **Windows** Virtual Machine by using an Azure Resource Manager **template** and Azure **PowerShell** from the Azure ***Cloud shell***
- The template used in this demo deploys a **single virtual machine** running Windows Server in a new **virtual network** with a **single subnet**
- For creating a Linux virtual machine, you can check here
- Creating an Azure virtual machine usually includes two steps:
  – Create a resource group
  – Create a virtual machine
- The following demo creates a VM from an Azure Quickstart template

edureka!



Data Privacy And Data Sovereignty In Azure

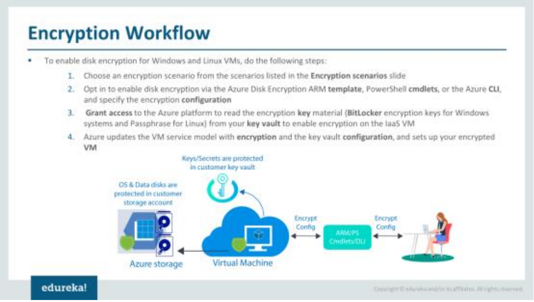
Azure enables you to control your **Azure-hosted data** through a range of advanced technologies to **encrypt**, **control** and **manage** encryption **keys**, and control and audit **access of data**

This control provides Azure customers with the ***flexibility*** to choose the solution that best meets their business needs

One such technology solution is "Azure Disk Encryption for Windows and Linux IaaS virtual machines (VMs)."

This technology helps **protect** and **safeguard** your data to meet your *organizational security* and *compliance commitments*

edureka!

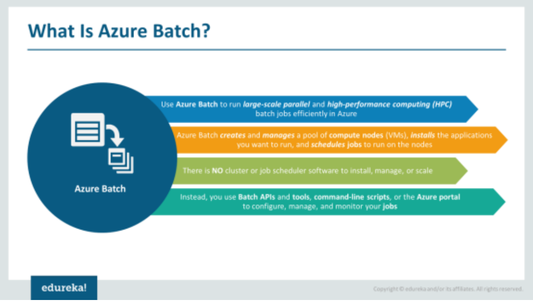


Encryption Workflow

- To enable disk encryption for Windows and Linux VMs, do the following steps:
  1. Choose an encryption scenario from the scenarios listed in the **Encryption scenarios** slide
  2. Opt in to enable disk encryption via the Azure Disk Encryption ARM **template**, PowerShell **cmdlets**, or the Azure **CLI**, and specify the encryption **configuration**
  3. **Grant access** to the Azure platform to read the encryption **key** material (**BitLocker** encryption keys for Windows systems and Passphrase for Linux) from your **key vault** to enable encryption on the IaaS VM
  4. Azure updates the VM service model with **encryption** and the key vault **configuration**, and sets up your encrypted **VM**

Keys/Secrets are protected in customer key vault
OS & Data disks are protected in customer storage account
Azure storage
Virtual Machine
Encrypt Config
ARM/PS Cmdlets/CLI
Encrypt Config

edureka!



What Is Azure Batch?

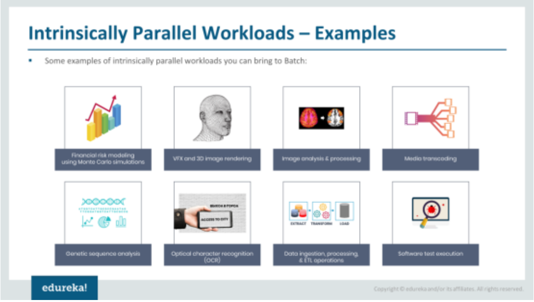
Use **Azure Batch** to run ***large-scale parallel*** and ***high-performance computing (HPC)*** batch jobs efficiently in Azure

Azure Batch ***creates*** and ***manages*** a pool of **compute nodes** (VMs), ***installs*** the applications you want to run, and ***schedules* jobs** to run on the nodes

There is **NO** cluster or job scheduler software to install, manage, or scale

Instead, you use **Batch APIs** and **tools**, **command-line scripts**, or the **Azure portal** to configure, manage, and monitor your **jobs**

edureka!



Intrinsically Parallel Workloads – Examples

- Some examples of intrinsically parallel workloads you can bring to Batch:

Financial risk modeling using Monte Carlo simulations

VFX and 3D image rendering

Image analysis & processing

Media transcoding

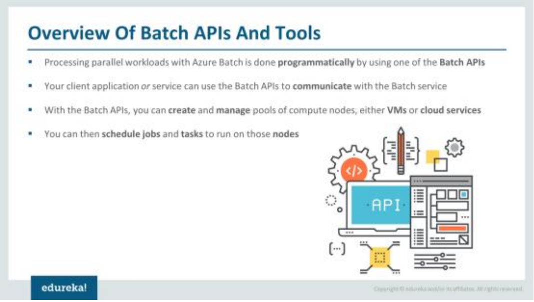
Genetic sequence analysis

Optical character recognition (OCR)

Data ingestion, processing, & ETL operations

Software test execution

edureka!



Overview Of Batch APIs And Tools

- Processing parallel workloads with Azure Batch is done **programmatically** by using one of the **Batch APIs**
- Your client application *or* service can use the Batch APIs to **communicate** with the Batch service
- With the Batch APIs, you can **create** and **manage** pools of compute nodes, either **VMs** or **cloud services**
- You can then **schedule jobs** and **tasks** to run on those **nodes**

edureka!

Questions

FEEDBACK

Comments

Ratings

Suggestions

Survey

Likes

Ideas

edureka!

Thank You

For more information please visit our website
www.edureka.co