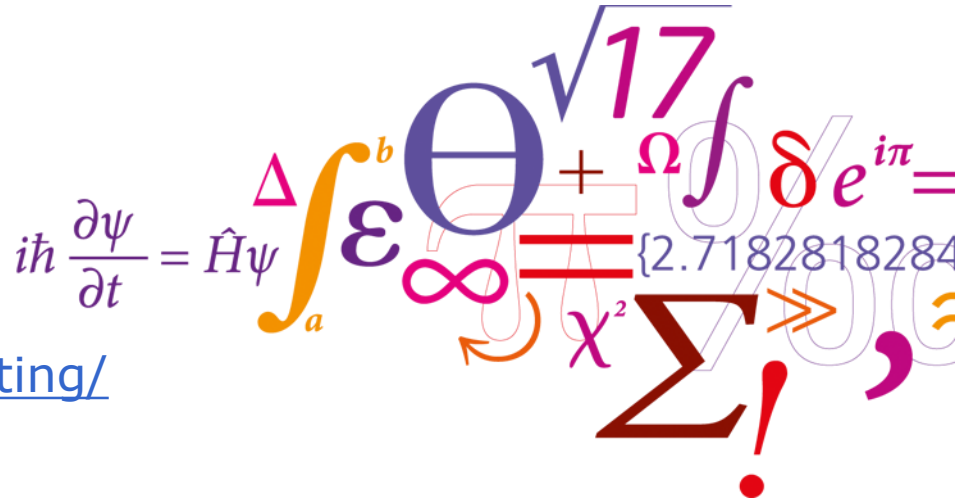
Pathfinding into the clouds

”Brug af Microsoft Azure i det lokale HPC anlæg”
eller: At finde vej op i skyerne

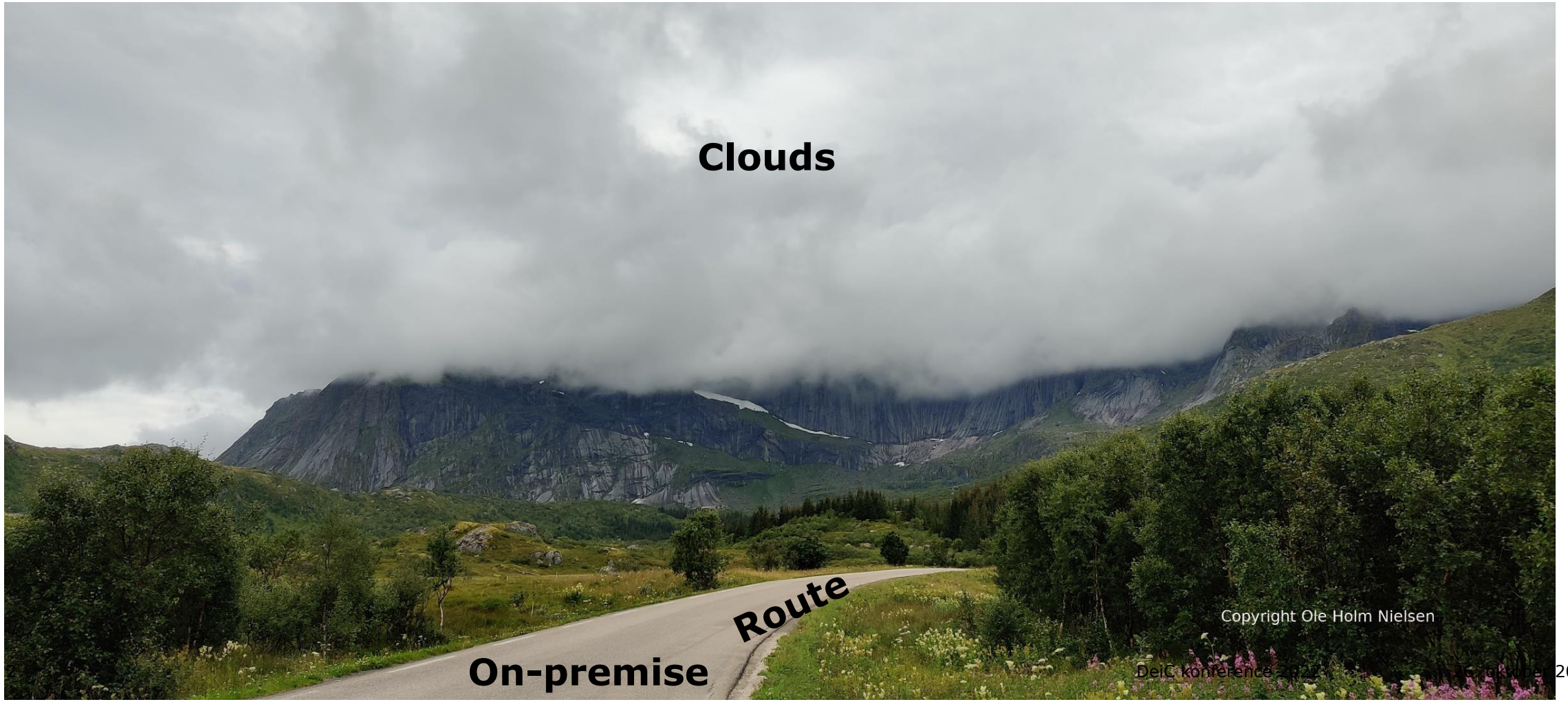
Ole Holm Nielsen
PhD, Senior HPC Officer
DTU Fysik



Henvisning til Wiki-siden:
https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/

DTU Fysik
Institut for Fysik

Pathfinding: Plotting by a computer application the shortest route between two points (Wikipedia)



Scaling an on-premise cluster into the clouds

- Exploring cloud technology for adding value to on-premise resources.
- **Note**: All cloud services charge a cost for server allocations, network traffic, and storage. Cloud costs are “running expenses” as opposed to “capital investments”.
- Extend local servers on-demand with **dynamically allocated servers** in the cloud. Servers are configured just like local servers, and user accounts are created the same way. No requirement of using the cloud’s sign-on solution (such as *Azure Active Directory*).
- **Seamless network integration** between on-premise and cloud. It’s just like getting new servers, except they are located on a private subnet in the cloud! Cloud servers should not be exposed to the Internet for security reasons!
- Get access to **CPU and GPU hardware types** different from what is available on-premise.
- Scale to a **large number of servers** for highly parallel jobs or spike loads.

Outline of this talk

Note: This work is based on the **Microsoft Azure** cloud service with guidance from Azure experts in Denmark.

1. Open an account at the cloud service provider.
2. Create cloud *Virtual Machines* (**VM**) and a *Virtual Network* (**Vnet**).
3. Create a **VPN IPsec tunnel** between your on-premise subnet and the cloud **Vnet**.
4. Write a script to power up/down VMs.
5. Define storage space in the cloud for application software and user data.
6. Work with the cloud nodes!

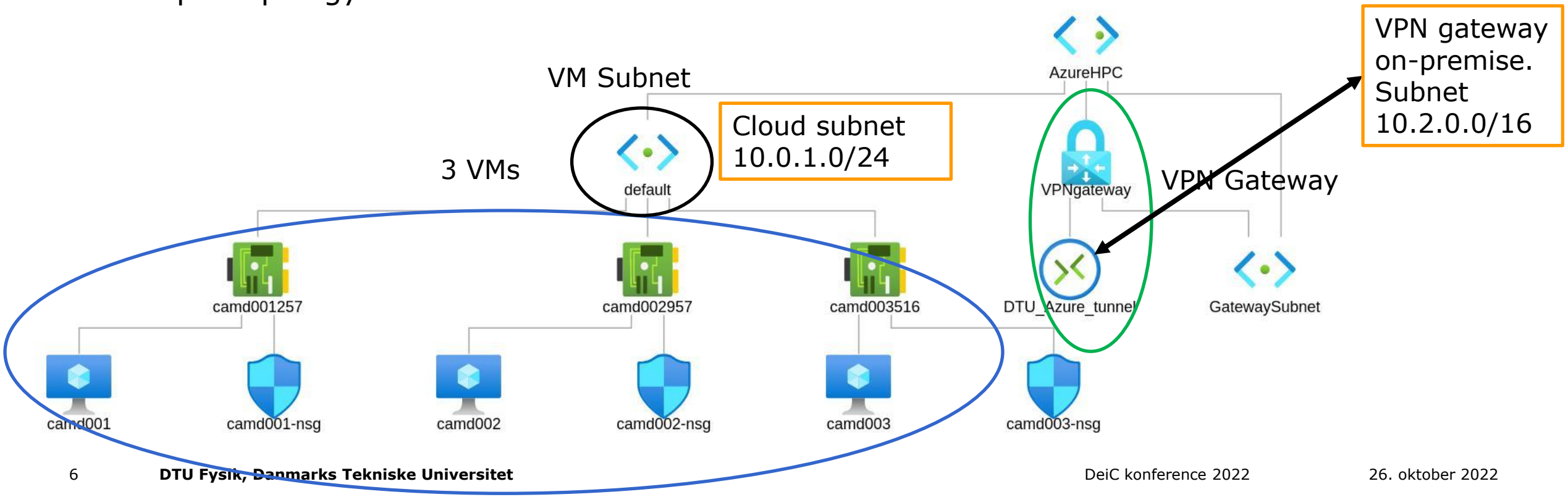
Open an account at a cloud service provider

- Many universities and other organizations already have a central IT **Azure Subscription**. We simply asked our central IT service to create an Azure account for us.

- Details are at https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#resources-for-slurm-on-microsoft-azure

Create cloud Virtual Network and Machines

- Create an Azure **Virtual Network** (**Vnet**)
  https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#virtual-network-in-azure
- Create an Azure **VPN Gateway**
  https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#vpn-gateway-to-azure
- Example topology for a **Vnet** named *AzureHPC:*



Create a Linux Virtual Machine

- Create a VM based upon some pre-existing minimal VM image: https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#create-resources-and-machines-in-azure-home
- **AlmaLinux** images are freely available in **Azure**.
- **RockyLinux** images are only available for pay in **Azure**.
- You must save the VM's *SSH public key file* and use it later to login via the *VPN tunnel*.
- **Remember** to shut down VMs when they are not in use!

On-premise IPsec VPN Gateway

- This is the least documented part of the cloud bursting adventure!
- Create a **Site-to-Site IPSec VPN connection** in Azure: https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#configure-vpn-gateways
- Azure provides a list of supported *compatible hardware router devices* (Cisco, Juniper, etc.) with links to vendors' configuration guides.
- **Problem**: What to do if you don't have the money to buy an expensive router, and/or the time to become an expert in the router's OS configuration???

Solution: Build your own IPsec VPN tunnel

Beware of strange problems lurking in tunnels

Libreswan comes to the rescue

- **Libreswan** is a free software implementation of the most widely supported and standardized VPN protocol using *IPsec* and the *Internet Key Exchange* (IKE).
- Use any Linux server with 2 NICs as the on-premise VPN gateway to the cloud service. One NIC faces the public Internet, the other NIC is in your cluster's private subnet (such as 10.2.0.0/16).
- *RHEL 8* (and clones such as *RockyLinux* and *AlmaLinux*) comes with *Libreswan* v4.4.
- There are numerous web-pages with instructions for setting up a site-to-site VPN tunnel. We have tried dozens of these methods, but none have worked on an EL8 server ☹
- Azure does not *support* Libreswan for VPN tunnels ☹
  But it works nevertheless!

A Libreswan setup that *actually works*!

- *Libreswan* setup details (firewall, IPsec, routing, etc.) are in: https://wiki.fysik.dtu.dk/it/Libreswan_IPsec_VPN

- Example IPsec configuration file `/etc/ipsec.d/azure.conf`:

```
conn azure                # Connection name
left=123.45.67.89         # Local VPN gateway public address
leftsubnet=10.2.0.0/16    # Local subnet
leftsourceip=10.2.0.1     # Local VPN gateway on the local private subnet
right=20.21.22.23         # Azure VPN gateway public address
rightsubnet=10.0.0.0/16 # Azure subnet
authby=secret             # Use shared secret with Azure
auto=start                # Start Ipsec at reboot
dpdaction=restart         # Restart if peer has died
dpddelay=30               # Dead peer delay
dpdtimeout=120            # Dead peer timeout
ike=aes256-sha1;modp1024 # IKE encryption/authentication algorithm
ikelifetime=3600s         # IKE renegotiation
pfs=yes                   # Perfect Forward Secrecy
esp=aes128-sha1           # Child SA negotiation algorithms
salifetime=3600s          # Expiry of a connection
```

Integrating cloud VM nodes with your on-premise cluster

- At this point hosts on your on-premise private subnet can communicate directly using TCP/IP via the *Libreswan IPsec* router (or a hardware router) to the Azure VNet private subnet. Example: `on-premise$ ping 10.0.1.2`

- Now you can SSH to any running VM server to configure the OS and install any necessary applications, as you would do with *any* on-premise server.

- We use *Ansible* for node configuration: https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#ansible-with-azure

- **Authentication**: Configure users in the VMs as you normally do in your on-premise servers (for example, add users to `/etc/passwd`).

- Your custom VM can now be cloned to create many identically configured VMs: https://docs.microsoft.com/en-us/azure/virtual-machines/capture-image-portal

Scripts to power up and down Azure VMs

- You can use scripts with *Azure CLI* commands to control VMs.
- For Slurm there is a *Power Saving Guide* https://slurm.schedmd.com/power_save.html
- Configure Slurm powering scripts:
  https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#slurm-configuration-for-cloud-nodes
- Slurm powering up/down scripts for Azure:
  https://github.com/OleHolmNielsen/Slurm_tools/tree/master/cloud
  The scripts can also be used from the command line without using Slurm.

Azure shared NFSv3 Storage Blobs

- It is important to create a storage space for your data in the cloud service.
- In February 2022 Azure started offering **NFSv3 Storage Blobs**: https://docs.microsoft.com/en-us/azure/storage/blobs/network-file-system-protocol-support
- This support provides Linux file system compatibility at **object storage scale and prices**. Linux clients can NFS-mount a container in **Blob storage** from an Azure *Virtual Machine* (VM) and also from **any on-premises computer**.
- First set up an Azure *Storage Account*: https://wiki.fysik.dtu.dk/Niflheim_system/Slurm_cloud_bursting/#azure-storage-accounts
- Now you can NFS-mount the Azure Blob storage container's IP-address/DNS-name. This works inside your VMs as well as in your on-premise servers (via the VPN tunnel)! We use the NFS auto-mounter with Azure Blob storage for ease-of-use.
- You probably want to create user home-directories in the Azure storage.

Slurm testing with cloud nodes

- Read the *Slurm Cloud Scheduling Guide*: https://slurm.schedmd.com/elastic_computing.html
- Add cloud nodes to Slurm.
- Submit batch jobs to run on the cloud nodes.
- Cloud nodes will be powered up and down on-demand by Slurm.

Conclusions

- Exploring cloud technology for adding value to on-premise compute resources.
- Cloud services such as Azure enable you to create a private cloud subnet (**VNet**) and populate it with **Virtual Machines** (VMs), choosing among many different types of hardware.
- Start with a standard Linux VM image and configure it for your needs.
- A **VPN IPsec tunnel** connects the cloud subnet to the on-premise private subnet.
- The **Libreswan** open-source software enables an inexpensive **IPsec VPN tunnel**.
- Azure offers inexpensive **NFSv3 Storage Blobs** for home-directories and data.
- CLI scripts are used to power up/down VMs, and are tightly integrated with the Slurm batch queue system.