

Moving Data Between Data Centers

1. Create a new network volume in the target data center.
2. Deploy two pods using the default RunPod PyTorch template, mounting the source and target network volumes separately.

The screenshot displays the 'Deploy GPU Pod' interface. At the top, there's a header 'Deploy GPU Pod' and links for 'Docs' and 'Ref'. Below this is a grid of GPU options:

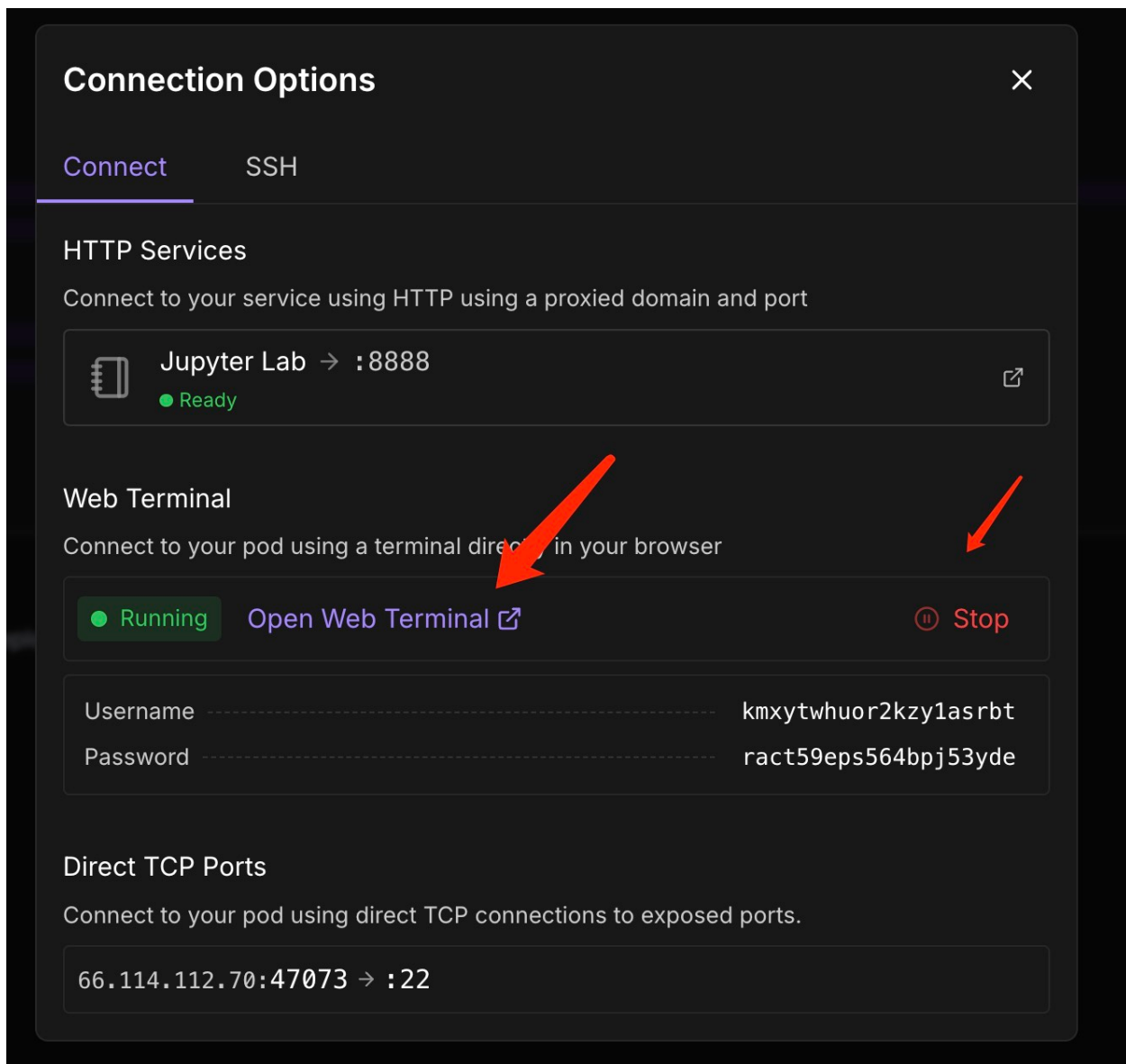
GPU Model	Price/hr	VRAM	CPU	Status
A100 SXM	\$1.89/hr	80 GB VRAM	250 GB RAM - 32 vCPU	Low
A100 PCIe	\$1.64/hr	80 GB VRAM	8 max	Unavailable
A40	\$0.44/hr	48 GB VRAM	10 max	Unavailable
RTX A6000	\$0.76/hr	48 GB VRAM	10 max	Unavailable
RTX 3090	\$0.43/hr	24 GB VRAM	6 max	Unavailable
RTX A5000	\$0.36/hr	24 GB VRAM	10 max	Unavailable
RTX A4500	\$0.34/hr	20 GB VRAM	4 max	Unavailable
RTX A4000	\$0.32/hr	16 GB VRAM	12 max	Unavailable

Below the grid, there's a 'Configure Deployment' section. The 'Pod Name' is 'source'. The 'Pod Template' is 'RunPod Pytorch 2.1' (highlighted with a blue border). The 'GPU Count' is set to 1. The 'Instance Pricing' section shows several plans:

Plan	Price/hr	Notes
On-Demand	\$0.69/hr	Pay as you go, with costs based on actual usage time.
1 Week Savings Plan	\$0.66/hr (\$110.88 total)	Reserve a GPU for one week at a discounted hourly cost.
1 Month Savings Plan	\$0.62/hr (\$416.64 total)	Reserve a GPU for a month at a discounted hourly cost.
3 Month Savings Plan	\$0.59/hr (\$1259.65 total)	Reserve a GPU for three months at a discounted hourly cost.
Spot	\$0.35/hr	Pay much less for an interruptible instance.

Additional options include 'Encrypt Volume' (unchecked), 'SSH Terminal Access' (checked), 'Start Jupyter Notebook' (checked), and 'Hugging Face' (Alpha).

3. Start the web terminal in both pods to begin the data transfer.



4. Install all packages in both source and destination servers

```
apt update && apt install -y vim && apt install -y rsync
```

5. Generate a pair of ssh key on **source** machine

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa -N "" -q && cat ~/.ssh/id_rsa.pub  
# Copy the public key
```

6. Copy the above public key and add to the destination machine
`authorized\_keys`

```
vi ~/.ssh/authorized_keys  
# Insert the public key
```

7. Take the destination Pod IP and Port number, Run the rsync command on
source machine

Connection Options

Connect SSH

HTTP Services
Connect to your service using HTTP using a proxied domain and port

Jupyter Lab → :8888
● Ready

Web Terminal
Connect to your pod using a terminal directly in your browser

● Stopped Start

Direct TCP Port
Connect to your pod using direct TCP connections to exposed ports.

66.114.112.70 34592 → :22

IP

Port

```
# sync files between source /workspace to destination /workspace  
# using destination machine's IP address and port number in below command
```

```
rsync -avz -e "ssh -p destination_port_number" /workspace/ root@destination_ip:  
# Example:  
rsync -avz -e "ssh -p 21345" /workspace/ root@149.7.4.139:/workspace
```