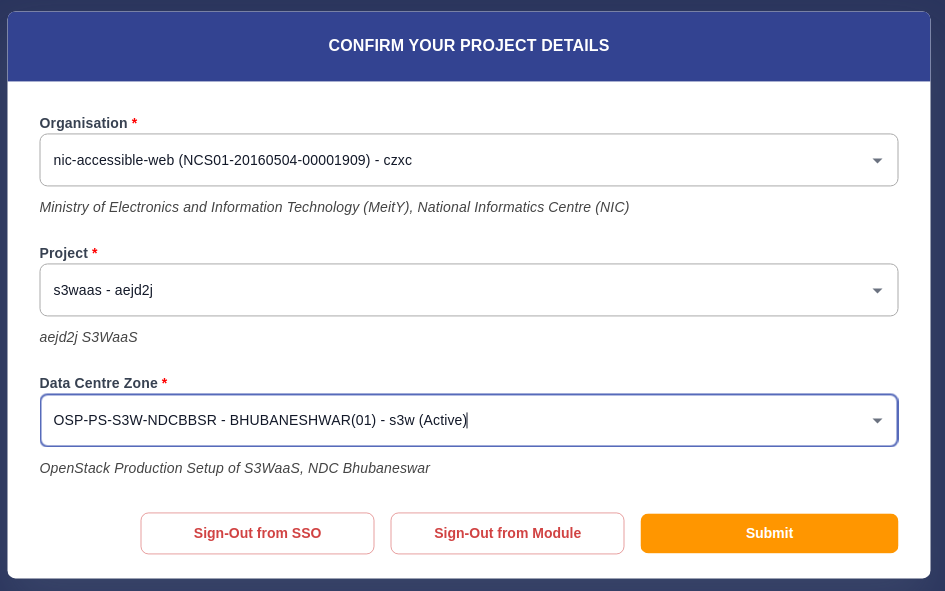
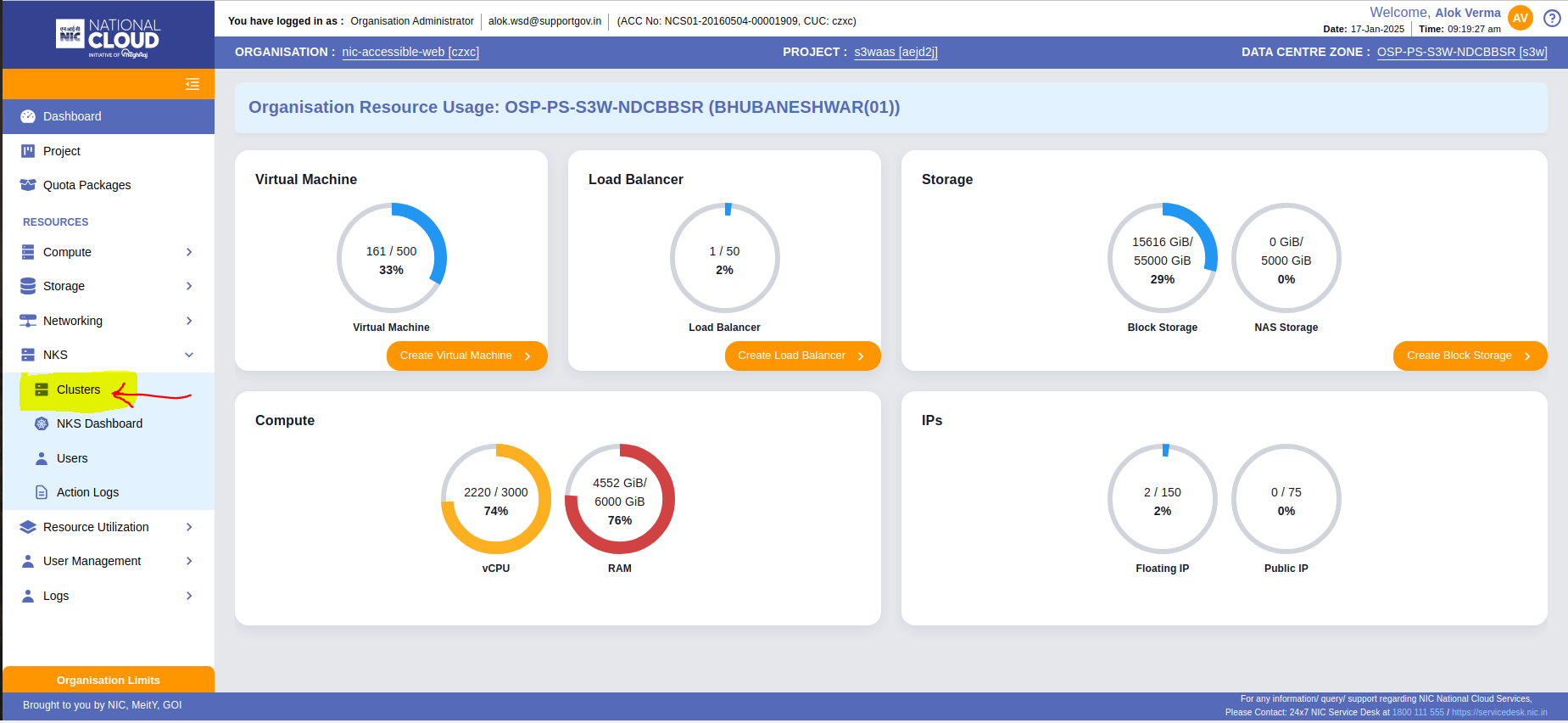
Creating Kubernetes Cluster on NKS platform and making it production ready.

1. Create Cluster

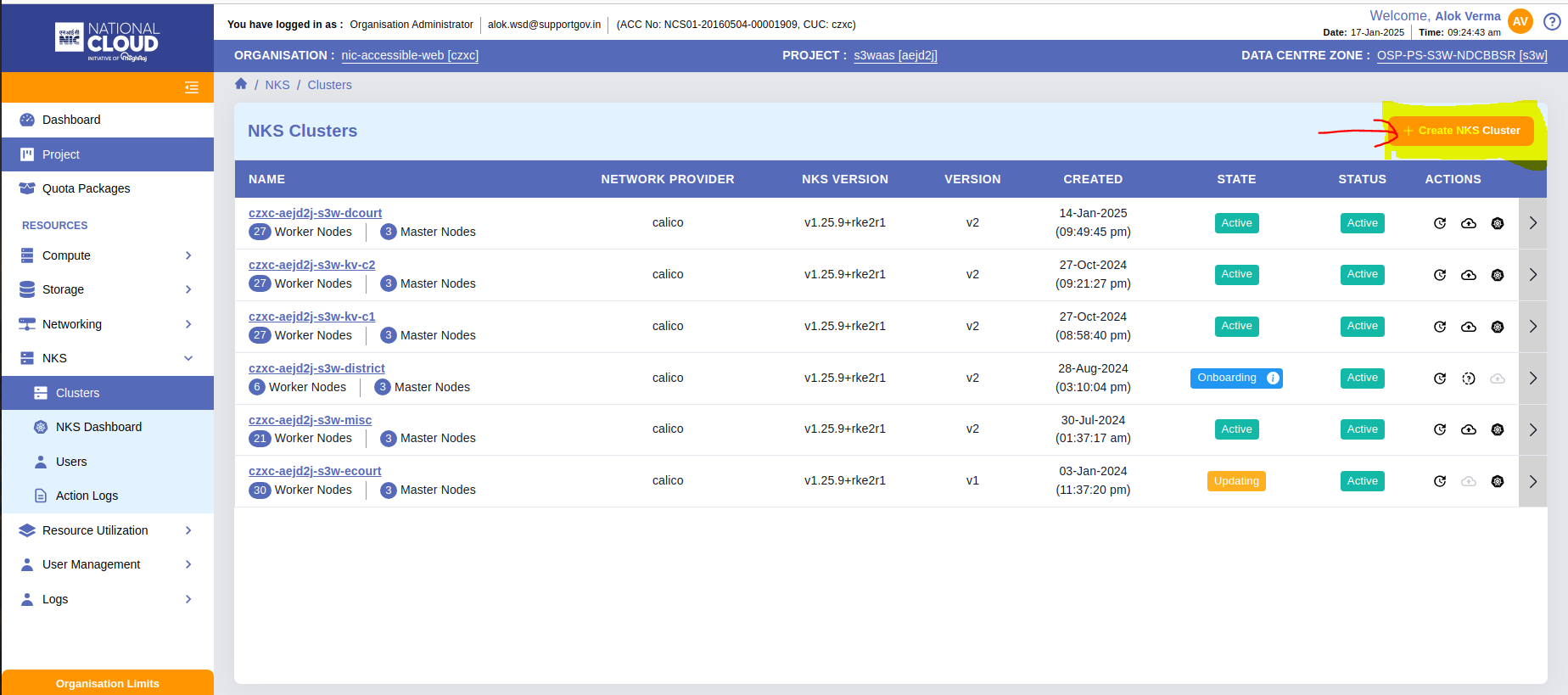
Login to <https://service.cloud.gov.in> portal. After successful authentication, select the given project details in project selection window.



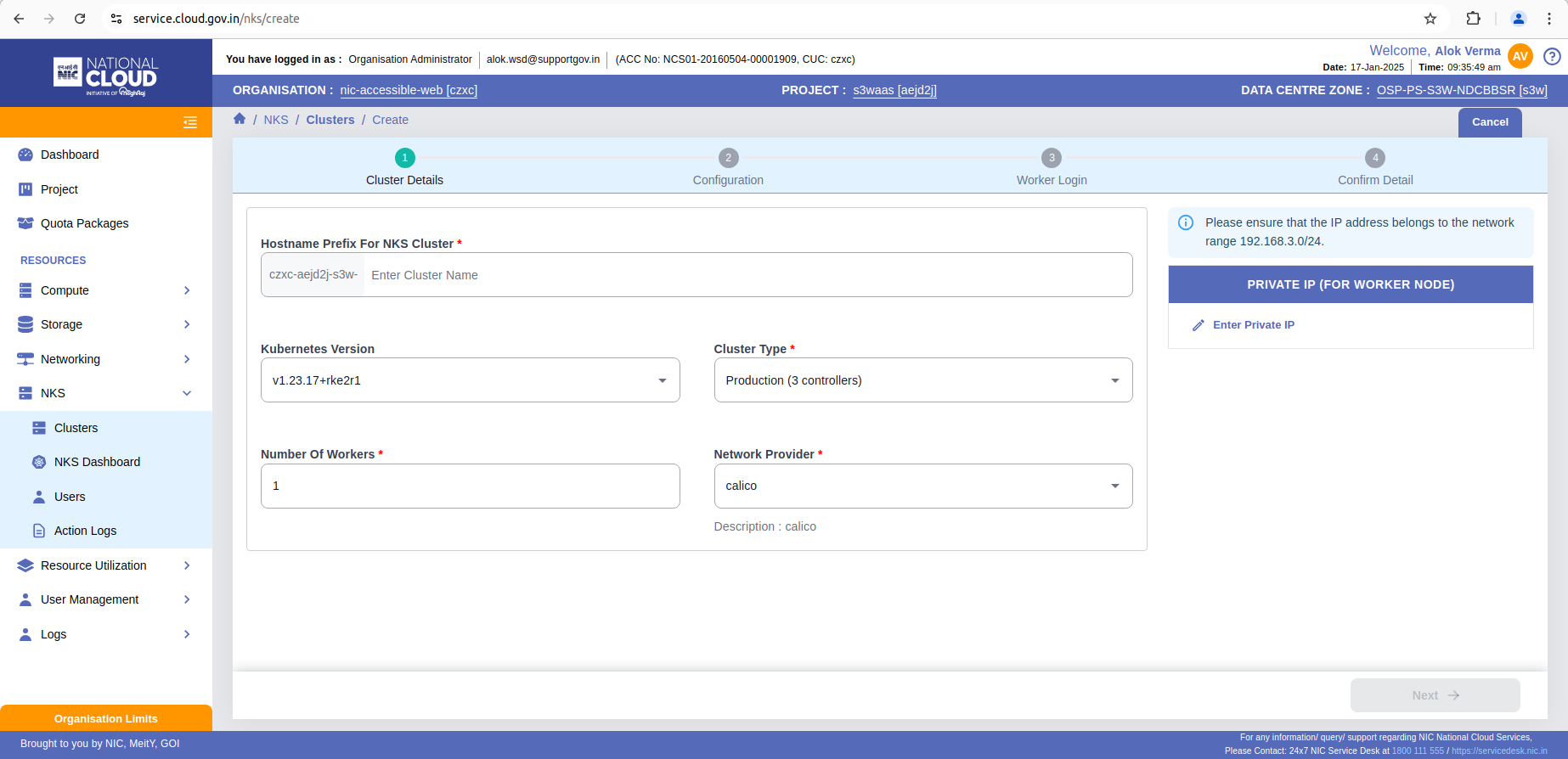
Now select the "Clusters" option from the NKS menu on the left sidebar.



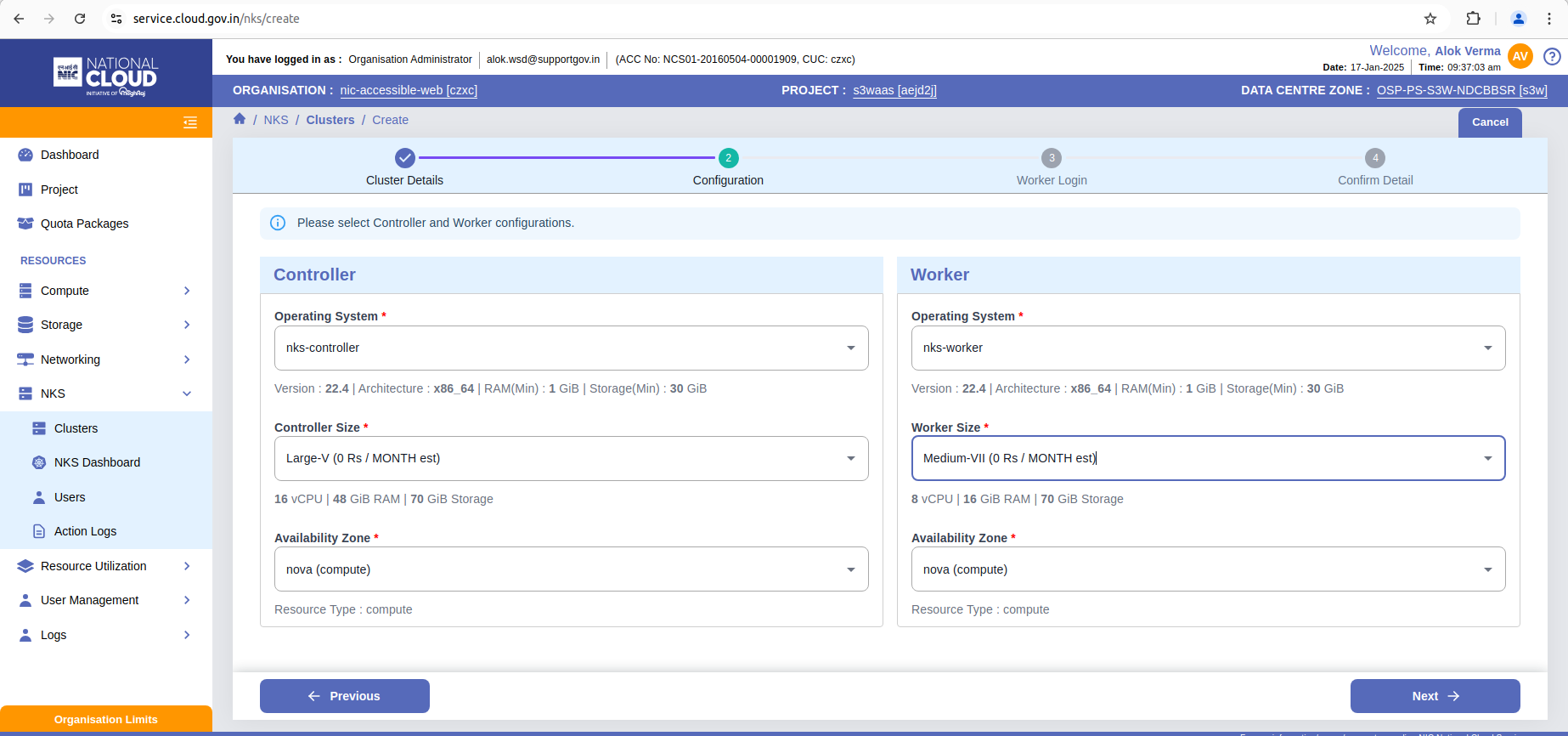
Click on "Create New Cluster" located in the top-right corner.



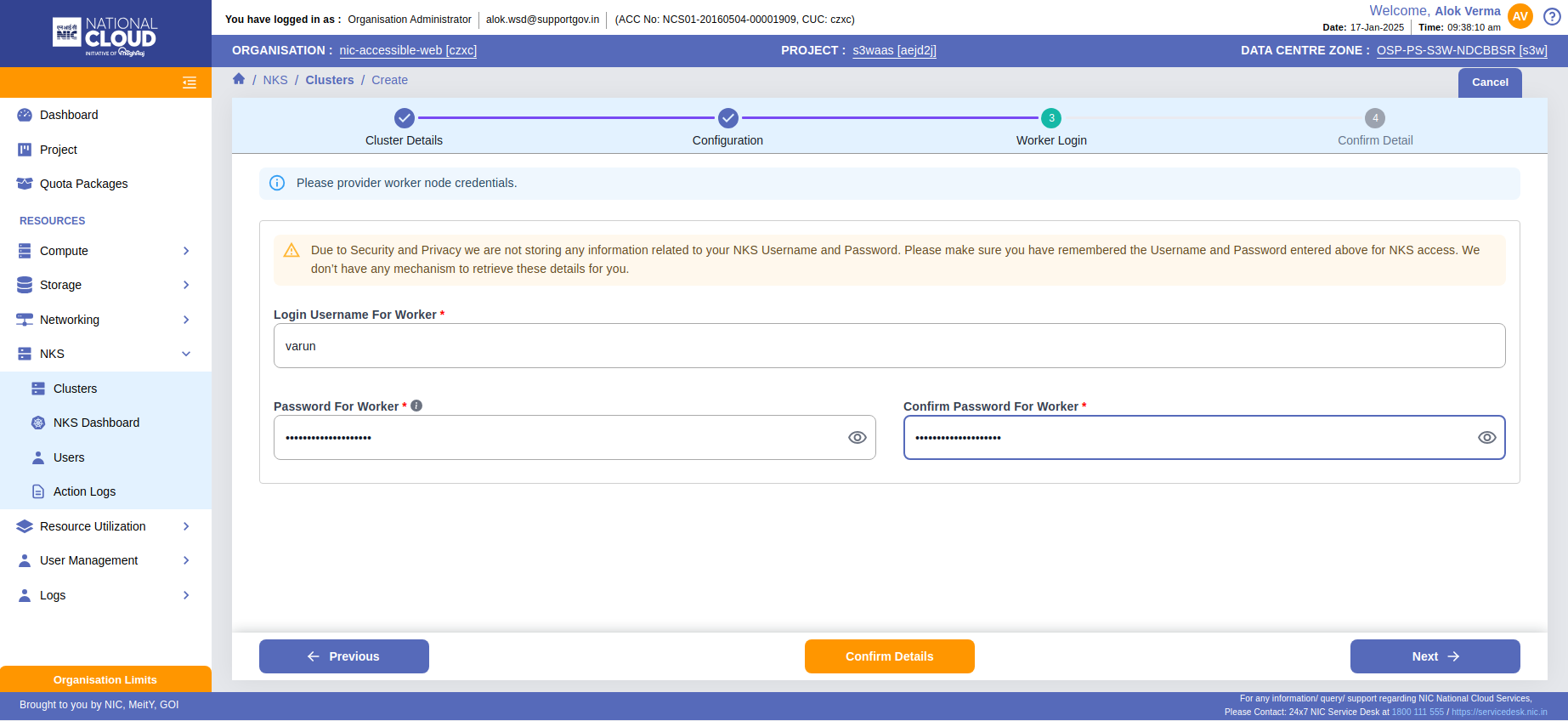
Provide the name of the cluster and select the desired Kubernetes version. Choose "Production" as the cluster type. Begin with a single worker node and select Calico as the network provider.



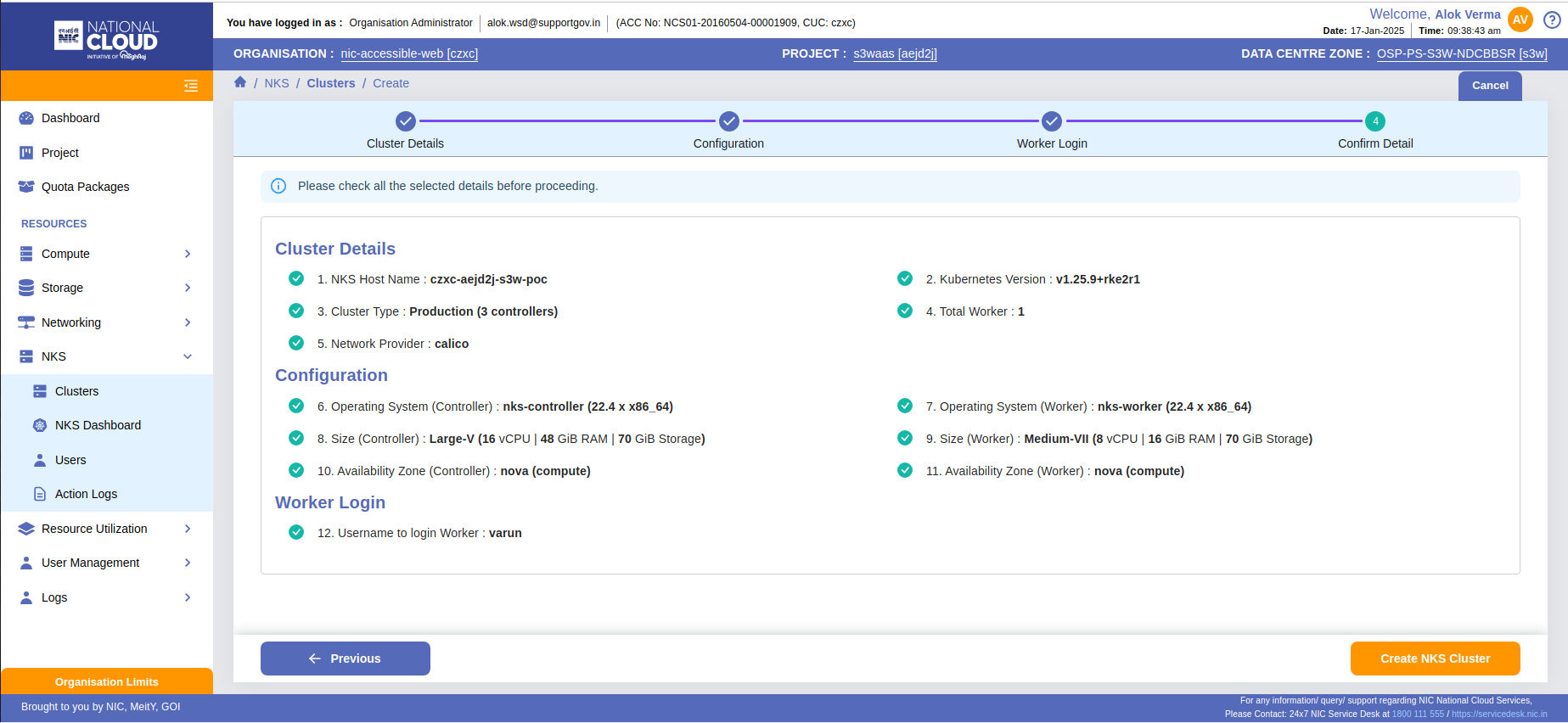
Choose nks-controller as the operating system for the controller and nks-worker for the worker nodes. Configure the size and availability zone based on your requirements.



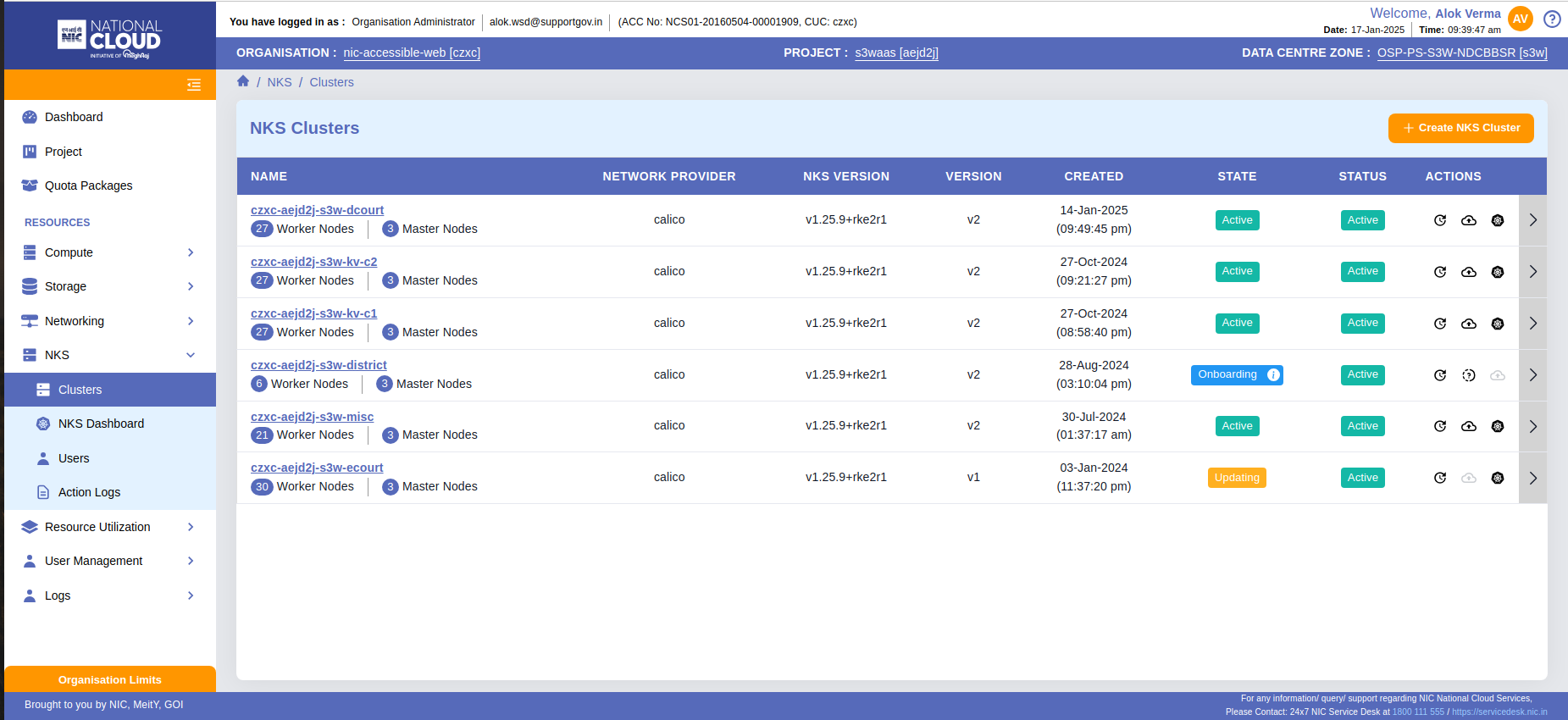
Provide a username and corresponding credentials that will be used for logging into the cluster nodes.



Review and confirm the details, then click the Create NKS Cluster button located at the bottom right.

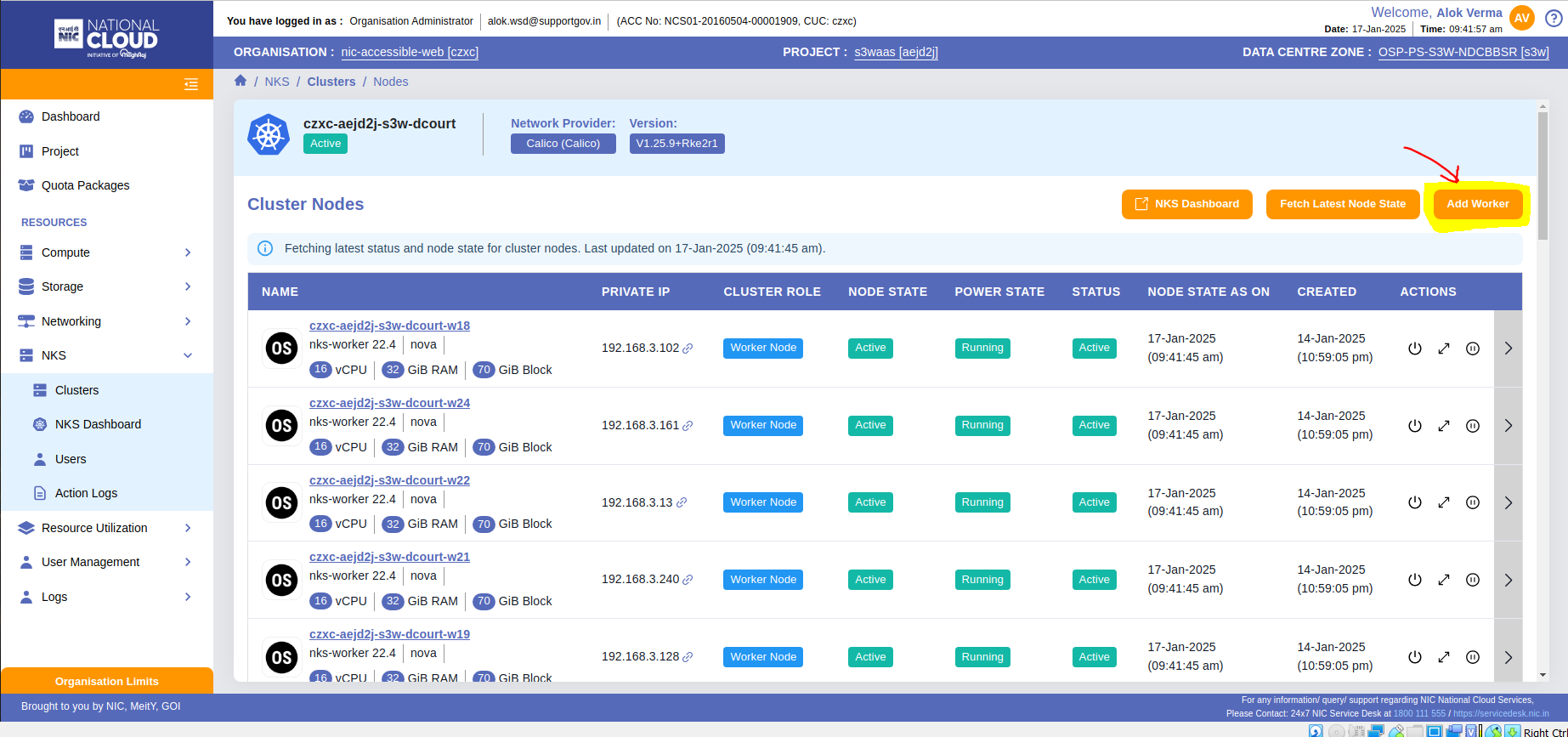


After submitting the request for cluster creation, the new cluster will appear in the list. Wait until the cluster's status changes to Active. Once the status is active, the cluster is ready for use.

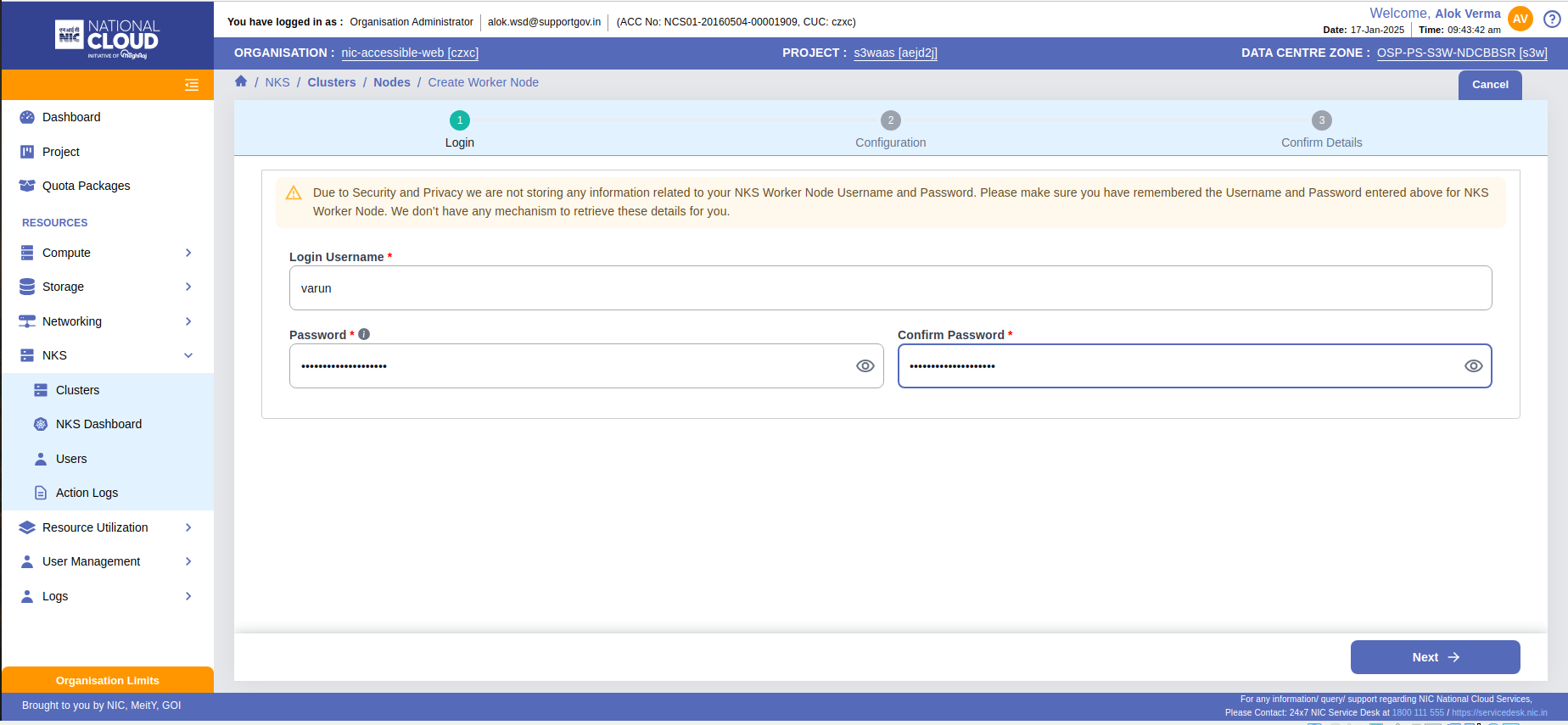


# Adding worker node to the cluster

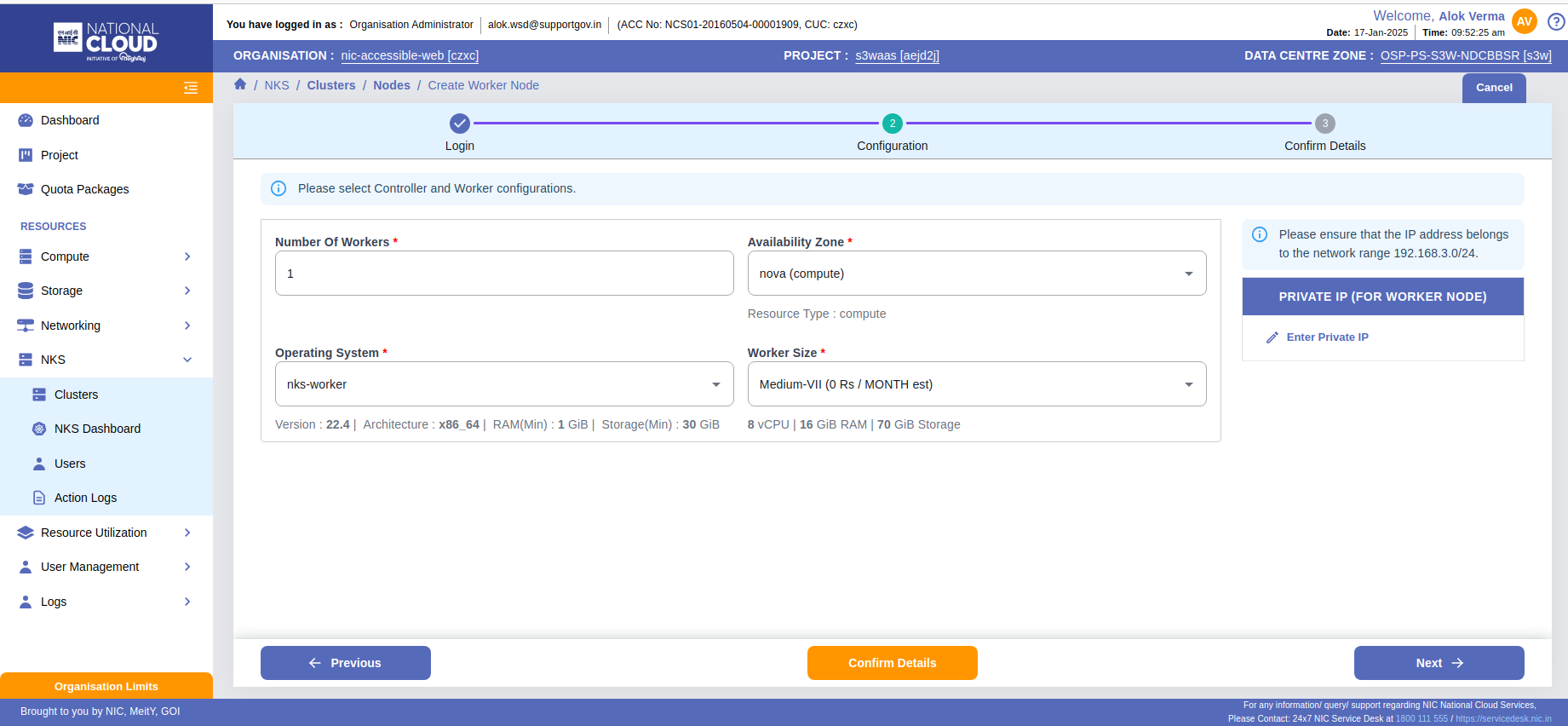
Select the cluster where you want to add a worker node, then click the Add Worker Node button located at the top right.



Provide a username and corresponding credentials that will be used for logging into the cluster nodes.



Specify the number of worker nodes, choose the availability zone, select the operating system, and configure the size for the worker nodes.



Review and confirm the details, then click the Add woker node button located at the bottom right.

**Here are the tasks to perform once the cluster is created:**

1. **Disable Swap and Reboot**:
   * Disable swap on all cluster master and worker nodes.
     + Run sudo swapoff -a on each node.
   * Edit /etc/fstab to comment out the swap line to persist across reboots.
   * Reboot the nodes: sudo reboot.
   * Verify swap is disabled with free -h or swapon -s.
2. **Set fs.inotify.max\_user\_instances**:
   * Create or edit the file /etc/sysctl.d/10-fs-inotify-max-user-instances.conf.
   * Add the following line:

fs.inotify.max\_user\_instances=400000

* + Apply the changes: sudo sysctl -p.

1. **Add FIP on Designated Ingress Worker Nodes**:
   * Assign a Floating IP (FIP) to the designated ingress worker nodes as per your requirement.
2. **Add Security Group for Ports 80-443**:
   * Add a security group rule to allow incoming traffic on ports 80 and 443 for the ingress worker nodes.
3. **Apply Taint on Ingress Worker Nodes**:
   * Run the following command to prevent user workloads from being scheduled on ingress nodes:

kubectl taint nodes <node-name> ingress=true:NoSchedule

1. **Add Toleration on Nginx-Ingress Pods**:
   * Edit the nginx-ingress deployment to include the toleration for ingress nodes:

tolerations:

- key: "ingress"

operator: "Equal"

value: "true"

effect: "NoSchedule"

1. **Add Storage Class to the Cluster**:
   * Define a storage class in your cluster based on your storage provider

kubectl apply -f <storage-class-file>.yaml

1. **Install rsync Package on Nodes**:
   * Run the following command on each node:

sudo apt-get install rsync

1. **Copy WAF and WP-Codes to Worker Nodes**:
   * Use rsync or scp to copy the WAF and WP codes from the source to the target worker nodes.
2. **Add to /etc/hosts**:
   * Add the following entry to the /etc/hosts file on each node:

192.168.3.69 registry.s3waas.gov.in